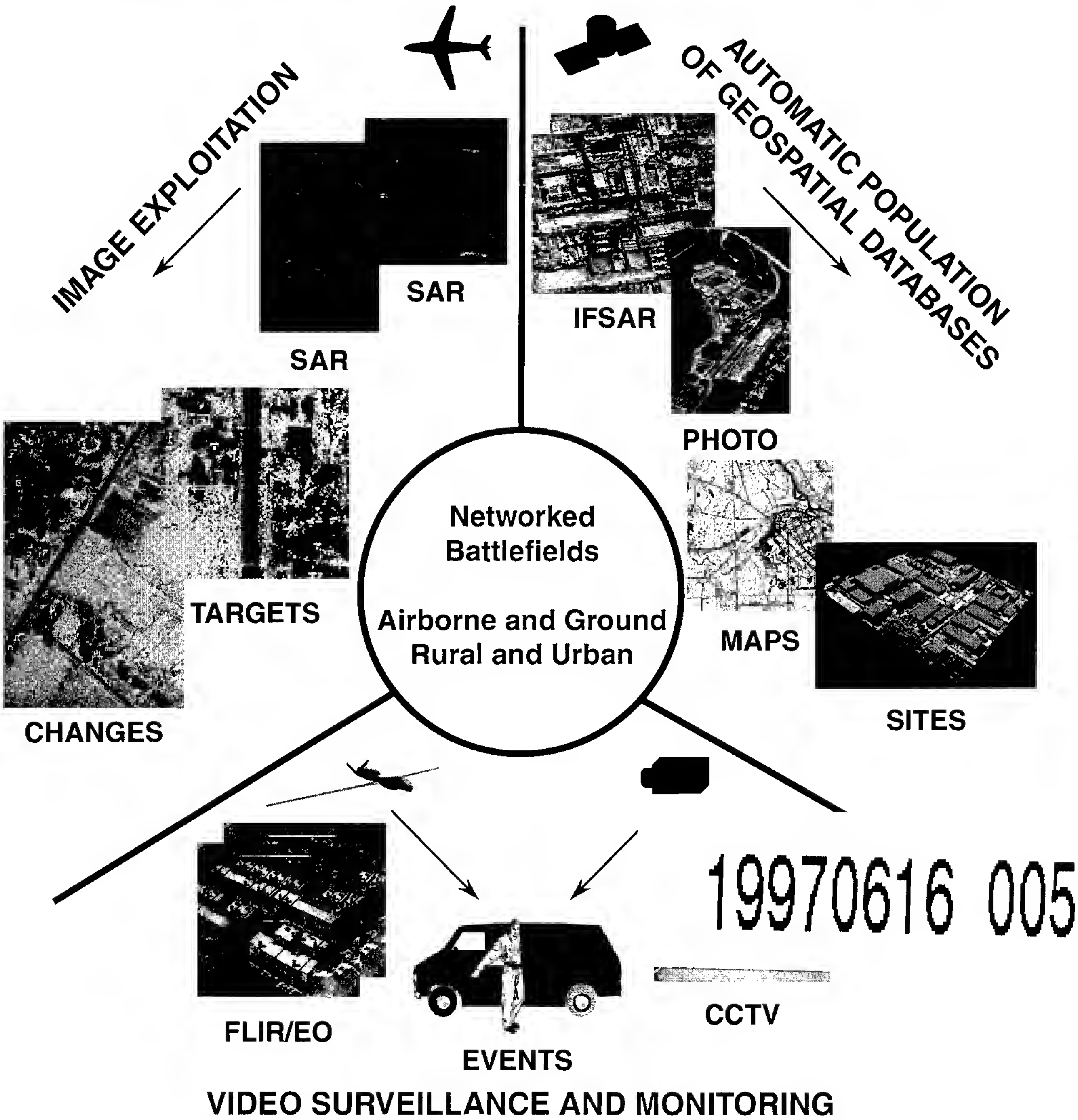
Proceedings
1997 Image Understanding Workshop
11 - 14 May
Hyatt Regency, New Orleans, LA

Volume II



Edited by
Thomas M. Strat

Sponsored by
Defense Advanced Research Projects Agency
Information Systems Office

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Proceedings, 1997 Image Understanding Workshop, 11-14 May, Hyatt Regency, New Orleans, LA

5. FUNDING NUMBERS

6. AUTHOR(S)
Thomas M. Strat, Image Understanding Program Manager

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

DARPA Information Systems Office
3701 N. Fairfax Drive
Arlington, VA 22203

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA Information Systems Office
3701 N. Fairfax Drive
Arlington, VA 22203

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This proceedings contains the assembled reports of the various research projects that comprise the DARPA Image Understanding (IU) Program. Submissions from forty academic and industrial computer vision research laboratories document progress and lessons learned in research performed for applications in image registration, target recognition, image exploitation, cartography, 3D model reconstruction, video surveillance, activity recognition and content-based image retrieval.

This is the 25th proceedings in the series, which has become known as a comprehensive source for the latest research results in image understanding from the nation's leading IU laboratories. Like its predecessors, this proceedings is not peer-reviewed in the traditional sense of a refereed conference or journal. Instead, the principal investigator of each laboratory is responsible for selecting the papers that will represent the work carried out in his lab. Because the reputation of each lab is at stake, the quality of submissions has remained consistently high.

14. SUBJECT TERMS

15. NUMBER OF PAGES
1531

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclass	Unclass	Unclass	

IMAGE UNDERSTANDING WORKSHOP

Proceedings of a Workshop

held in

New Orleans, Louisiana

May 11 - 14, 1997

Volume II

Sponsored by:

Defense Advanced Research Projects Agency
Information Systems Office

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are Principal Investigator reports and technical results from both the basic and strategic computing programs within DARPA/ISO sponsored projects and certain technical reports from selected scientists from other organizations.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Government of the United States of America.

Distributed by:
Morgan Kaufmann Publishers, Inc.
340 Pine Street 6th Floor
San Francisco, CA 94104-3205
ISBN: 1-55860-490-1
Printed in the United States of America

Printed by: ***Omnipress*** *. . . . "Helping Associations Educate"*
Madison, Wisconsin (800) 828-0305

Table of Contents

Volume I

Section I: Video Surveillance and Monitoring (VSAM)

Video Surveillance and Monitoring - Principal Investigator Reports

Video Surveillance and Monitoring - Technical Papers

SECTION II: IMAGE EXPLOITATION (IMEX)

Image Exploitation - Principal Investigator Reports

Image Exploitation - Technical Papers

Section III: Image Understanding Environment (IUE)

Image Understanding Environment Technical Papers

Volume II

Section IV: Automatic Population of Geospatial Databases (APGD)

Automatic Population of Geospatial Databases - Principal Investigator Reports

Automatic Population of Geospatial Databases - Technical Papers

SECTION V: AUTOMATIC TARGET RECOGNITION (ATR)

Automatic Target Recognition - Principal Investigator Reports

Automatic Target Recognition - Technical Papers

Section VI: Multidisciplinary University Research Initiative (MURI)

Multidisciplinary University Research Initiative - Principal Investigator Reports

Multidisciplinary University Research Initiative - Technical Papers

Author Index

SECTION IV
AUTOMATIC POPLUATION OF GEOSPATIAL DATABASES
(APGD)

AUTOMATIC POPLUATION OF GEOSPATIAL DATABASES (APGD)

PRINCIPAL INVESTIGATOR REPORTS

An Integrated Feasibility Demonstration for Automatic Population of Geospatial Databases *

Martin A. Fischler, Robert C. Bolles, and Aaron J. Heller
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA
E-MAIL: {fischler,bolles,heller}@ai.sri.com
PROJECT-PAGE: http://www.ai.sri.com/~apgd

Abstract

The objective of this Automatic Population of Geospatial Databases (APGD) Integrated Feasibility Demonstration (IFD) effort is to develop, evaluate, and demonstrate the enabling technology that will permit rapid, robust, automated population of geospatial databases, from a variety of imagery sources, to serve synthetic environments and imagery exploitation applications. SRI International and its team members GDE and Vexcel expect to provide procedures that will radically reduce the need for human intervention in the extraction of 3-D cartographic features and their attributes from imagery and supporting auxiliary data. The SRI team will concentrate on the features that are the most useful for the above target applications and the most time-consuming for a user to extract manually, such as roads, rivers, communication lines, buildings, and land cover. We plan to extend our "context-based vision" paradigm to provide a flexible architecture for incorporating and appropriately applying existing and future highly competent, but specialized feature extraction algorithms. For evaluation purposes and in support of the FRE contractors, we will construct one or more "ground truth" datasets, run independent benchmarks, and provide a mechanism for testing a new module within the latest version of the full APGD-IFD platform.

1 Introduction

Current practice in populating geospatial databases requires human involvement in almost all phases of the effort. Even though the IU community has produced a large body of effective scene analysis techniques, there are few (if any) completely automated and robust end-to-end processes for extracting any of the 50 to 200 features of probable interest. However, we believe that revolutionary gains in productivity *will* be possible in the near term, by creating an architecture that automates the "core" operations of the feature extraction process: selection and initialization of algorithms, selection of their parameters, and evaluation of the results.

A critical problem is that most of the currently available fully automated algorithms were developed and tested in restricted domains, and consequently will probably fail if their implicit assumptions are not satisfied. Furthermore, many of these assumptions and domain limitations are unknown to the algorithm developers themselves, let alone the application-oriented user.

Researchers at SRI International have developed an approach called *context-based vision* [Strat and Fischler, 1991], in which a collection of specialized algorithms is assembled to perform various extraction tasks on different types of data. Each algorithm is accompanied by a body of information that makes explicit as much as is known about the conditions under which the algorithm-parameter-image combinations will (or will not) work. Given a specific task

*This work was sponsored by the Defense Advanced Research Projects Agency under contract NMA100-97-C-5013 monitored by the National Imagery and Mapping Agency (SRAR), Bethesda, MD. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, or SRI International.

to be performed, the system selects the appropriate algorithms to be applied and the parameters to be used.

Our system for APGD is called the BOS (Battlespace Observer System). The scenario we envision is one in which a small amount of task specification and initialization is performed by a human operator. The system then runs unattended to populate, and continuously update, the database with the extracted features. Some human inspection and editing will probably be required. The nominal mode of operation is one of enhancement/intensification/update/confirmation of the database where there already exists some prior (perhaps crude) model. Cold-start will be possible because of the facilities provided by the underlying RCDE [Heller and Quam, 1997], but this will require an increased amount of human interaction. Somewhat paradoxically, it is clear that reducing human interaction requires well-engineered (and perhaps extensive) facilities for those tasks the human must perform to support the automated functions. The RCDE/HUB system has been designed to allow effective human interaction with automated and semiautomated algorithms to accomplish the task of constructing 3-D scene models from images, sensor data, and stored knowledge.

A major expected result of this APGD project will be a revolutionary reduction in the effort and time needed to construct 3-D attributed scene models, thereby making feasible widespread use of already-proven simulation, analysis, and visualization techniques, and enabling new applications. For example, new intelligence products (e.g., hyperlink-annotated VRML models, texture mapped with current imagery that can be retrieved via the World-Wide Web (WWW) or Intelink, or as an enabling technology for implementing "intelligent push" for battlefield dissemination of tactically relevant images that have been identified by an automatic model-based "Quick-Look" process.

1.1 Approach

Our approach includes the following components:

- **An overall modular and expandable system architecture** – called the BOS (for the Battlespace Observer System, see Figure 1) – based on a rigorous photogrammetric/geometric framework, that can accept inputs from a wide variety of sensor and other information sources, and is able to support a large number of distinct Battlefield Awareness applications (in particular, those concerned with constructing synthetic environments and model-supported exploitation). Using the current RADIUS/RCDE as a base, the proposed architecture will provide facilities for manual interaction where required (especially for initialization and editing) and the cartographic infrastructure to support automated algorithms without requiring them to provide their own machinery to interact with a geo-referenced model.

- **A context-based algorithm control system** (CBACS) that selects and parameterizes algorithms to assure their effective use. CBACS provides a modular way of integrating a large number of existing feature extraction techniques, that work well in narrow/specific situations, into a feature extraction and attribution system that is reliable over the broad range of conditions that will be encountered in real-world environments.

- **A sensor calibration and control subsystem** that supports cross-sensor analysis of NTM, SAR, IFSAR, IR, panchromatic, and hyperspectral imagery. This subsystem includes facilities for photogrammetrically rigorous error analysis and propagation, efficient procedures for applying image-to-world and world-to-image coordinate system transformations, and a uniform Application Programming Interface (API) for all sensor types.

- **A collection of algorithms for feature extraction and attribution.** *Feature Extraction Managers* will control the extraction process for each class and type of feature, including:

  1. linear features (roads, drainage, power lines)
  2. compact 3-D structures (buildings, power poles)
  3. aerial features (water bodies, land cover classification)

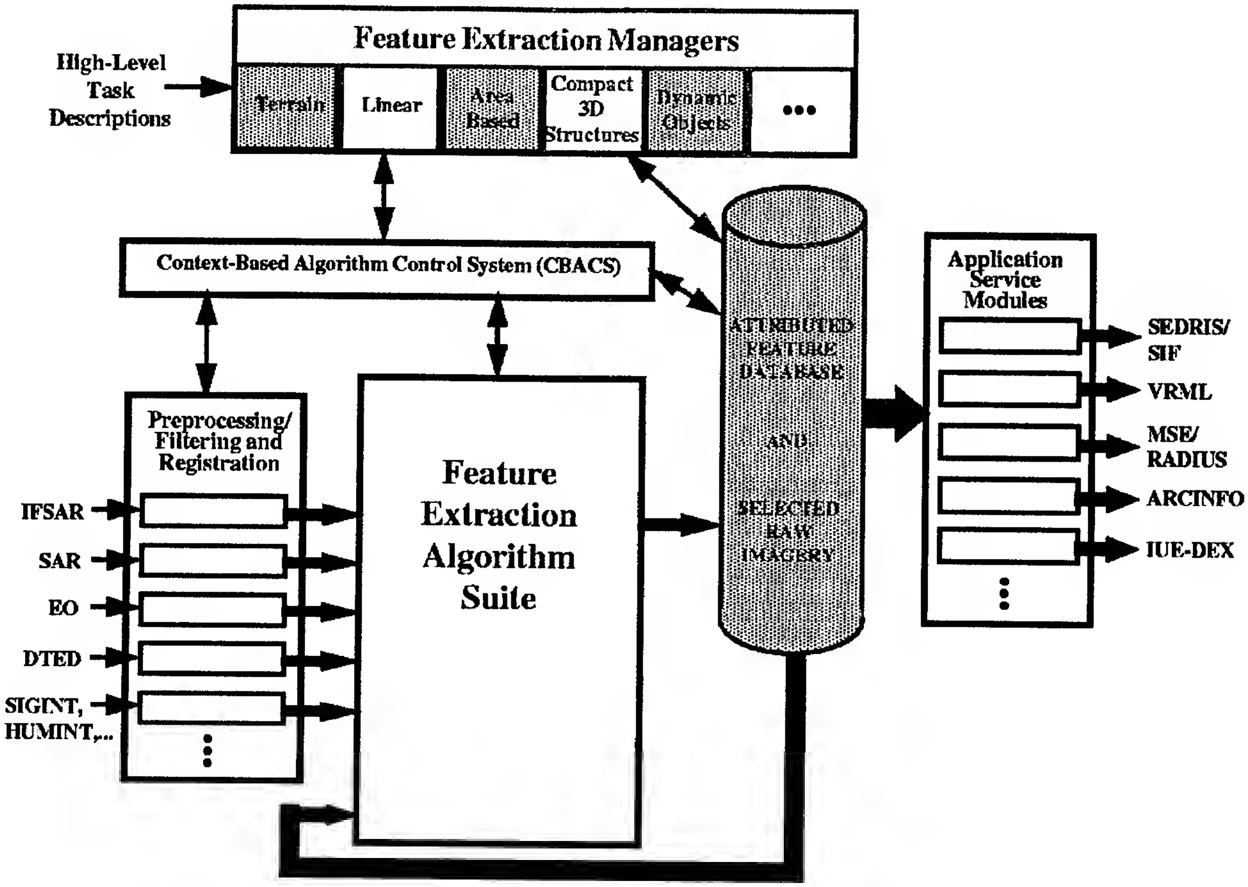

Feature Extraction Managers
High-Level Task Descriptions
Terrain
Linear
Area Based
Compact 3D Structures
Dynamic Objects
•••
Context-Based Algorithm Control System (CBACS)
Preprocessing/ Filtering and Registration
IFSAR
SAR
EO
DTED
SIGINT, HUMINT,...
Feature Extraction Algorithm Suite
ATTRIBUTED FEATURE DATABASE AND SELECTED RAW IMAGERY
Application Service Modules
SEDRIS/ SIF
VRML
MSE/ RADIUS
ARCINFO
IUE-DEX

Figure 1: The Battlespace Observer System Architecture

4. terrain (elevation, prominent visible features, mobility characteristics)
5. man-made movable objects (vehicles, time critical mobile targets)

- **A persistent object-oriented blackboard** to store our continuously updatable world model. It can accommodate incomplete and conflicting information – with pointers back to the relevant imagery when it is desirable for the application to present the user with a synthesized view of the scene rather than an interpreted description; this is especially important when the feature extraction algorithm knows it has failed to make a correct interpretation.

- **Establishment of an APGD "Virtual Lab,"** utilizing the WWW, for allowing continuous evaluation of our progress and for forming and strengthing an APGD community. The Virtual Lab will enable continuous access to raw data, contextual information, ground truth, extracted models, and evaluation procedures.

The major deliverable at the end of the two-year base period will be a detailed specification for a modular, extensible architecture, supporting algorithms, and evaluation techniques for the task of APGD. At the end of the full five-year contract, we will have produced an application-ready, end-to-end, prototype system that can be demonstrated and evaluated within an instrumented environment. This system will be suitable for quasi-operational use at TEC, the NEL, or other government labs. In the interim, we expect that many of our advances will have been transfered into operational systems.

2 Scientific Issues and Research Questions

The central problem we address in this effort is that of reducing the role of (or completely removing) the human operator from the "low-level" processes of constructing geospatial models from images and stored data. The primary contributions of the human operator are (1) to understand the nature of the task being performed so as to be able to evaluate the intermediate and final products of the automated functions, and (2) to select the appropriate extraction algorithms to be employed and to guide the extraction system toward a satisfactory answer by making appropriate adjustments in its parameters.

Thus, to remove the human, we must be able to produce an "objective function" (which we will call *ObjF*) that can numerically or categorically evaluate the adequacy (for some specified application) of a proposed interpretation of the visual data in the absence of explicit knowledge of "ground truth." If we can accomplish this goal, we can also accomplish the difficult part of step (2) by iteratively modifying the extraction-algorithm parameters until we achieve a suitable score for the suggested interpretation.

It is almost certainly the case that the amount of intelligence we would have to encapsulate in the *ObjF* to completely remove the human is not achievable in the relatively short time frame of this effort – but if we can adequately characterize the expected performance of the algorithms currently available, and if we can automatically evaluate their (the algorithm's) true likelihood of having produced a correct answer in any given trial, then we can indeed achieve the radical improvements in productivity we are expecting to come out of this work. The human would still have a critical role in initializing the system and in editing the final result/output, but he would be removed from the most time consuming tasks that he must now perform, and he would not have to understand the workings of the machine vision system itself, but only the application and its requirements.

The architecture we have suggested provides a framework for taking advantage of the proposed automation of the technique invocation and evaluation role of the human operator, it does not provide that technology in itself. It is clear that developing effective techniques for the prediction and evaluation of the results of the algorithmic (extraction) processes will be the key factor in the success of this work. Some of the major scientific problems to be addressed include:

2.1 (Automated) Evaluation

As discussed above, self-evaluation by an algorithm, or by the more comprehensive system controlling the algorithm, is perhaps the most important ingredient in achieving automation and robustness. We consider this problem our main scientific challenge in this effort.

In addition to (internal, on-line, real-time) self-evaluation, the more conventional problem of (external) objective evaluation is also critical to the success of our work. In order to properly evaluate performance, one should have a comprehensive model of the process to be evaluated and the problem domain on which the process operates, as well as definitive way to measure the correctness of a proposed answer. None of these items are currently available in the APGD domain. One of the most serious problems is that of obtaining adequate ground truth to do statistically meaningful testing. The obvious way to get such ground truth is to have the type of system we are proposing to develop (but don't yet have).

2.2 Robustness

The human operator is able to quickly and easily detect and remove gross errors (blunders) in algorithm performance. Since we can't hope to duplicate the human operators intelligence, commonsense, or experience, we must develop computationally effective procedures for combining independent automatically derived judgments on all important automated decision processes. Even if we are successful in detecting and removing most blunders, a few will undoubtily slip through our filters – as well as a larger number of almost correct but still erroneous numerical evaluations. Our compiled models and our data base will contain incomplete, uncertain, and conflicting information. Nevertheless, our composite system will be expected to perform consistently and correctly.

2.3 Introduction of an Automated Learning Component

The problem of automatically deriving optimal algorithm parameters based on scene content and acquisition conditions is not a realistic near term objective. It will probably be necessary to acquire this information through direct observation and evaluation of algorithm performance. We will develop and incorporate (into CBACS) an automated learning component to perform this function.

2.4 Algorithm Competence

Many of the existing extraction techniques we expect to use appear to be competent to perform their intended functions, but are deficient in some critical way; for example, they require exponentially increasing resources with increase in image size, or they make occasional blunders which they cannot detect (no self-evaluation), or they require some contextual information that might not be routinely available. The BOS architecture we are developing has machinery to cover some of these deficiencies, but the actual interface between a defective algorithm and the architecture may require innovation equal to that of directly improving the algorithm. The Architecture increases both our options and our chances for finding a fix; it does not eliminate the problem of producing more competent algorithms.

3 Evaluation

Our evaluation plan focuses on three high-level metrics associated with the automatic population of geospatial databases. The first is the amount of time a user requires to interact with the system to extract and assign attributes to a specified set of features. The second is the accuracy of the extracted features and their attributes. The third is the utility of the image sequences generated from the extracted models (to support "virtual worlds" applications).

Within the RADIUS program, we partially instrumented the RCDE system to monitor and record the actions of a user applying various techniques to extract roads [Heller *et al.*, 1996]. The system recorded the number of mouse-clicks, the selected actions, and the amount of mouse-travel required to achieve a desired result. We feel that this is a better measure than, for example, actual computation times because it truly reflects the amount of human interaction and does not depend on the speed of the computer being used.[1]

For this project, we plan to extend this approach and evaluate algorithms in terms of the type and amount of parameter tuning they need to perform a task and the types of errors they make. In this way, the performance of the algorithm can be parameterized by the level of human interaction required, both for initialization and clean-up, in addition to the accuracy of the result. This allows us to derive a *multi-dimensional* rating for an algorithm, rather than a simple rank ordering. To choose a particular algorithm for a given task in a given context, the requirements (e.g. time, accuracy, other available data) are used to weight the various ratings for an algorithm to derive an application-specific ranking.

To measure the accuracy of extracted features and their attributes, we plan to use our current tools to construct detailed models of three sites and compare them to models constructed with candidate automated feature extraction techniques. Figure 2 shows part of a model being constructed of the motor pool area at Ft. Hood. Portions of these "ground-truth" models will be available continuously on our web site, so FRE contractors and other groups can compare their latest results to ground truth. In addition, periodically, we will run evaluations using previously unreleased data, or new portions of the ground-truth models, to record and report progress on tasks where "algorithm tuning" is not possible.

The third metric is the utility of the dynamic visualizations or image sequences generated from the database. This is the most difficult attribute to quantify because it not only depends on the quantity and quality of the extracted features, but also on the task being performed (e.g., mission rehearsal) and on the rendering techniques and graphics engines used. We plan to explore a variety of approaches to this problem, including ideas such as asking users to answer questions about the site after viewing a "fly-through." This approach will provide an empirical evaluation of the generated sequences for specific tasks. We will also show sequences to people and ask them to list perceptual problems. For example, the inter-object visibility may not be handled correctly because the objects were not delineated properly, or the buildings may appear to float in the air because a consistency mechanism failed.

An important component of our evaluation and demonstration plans is the APGD Virtual Lab. Uti-

[1] Additional details can be found at URL: `http://www.ai.sri.com/~radius/sri/baa-reports/`.

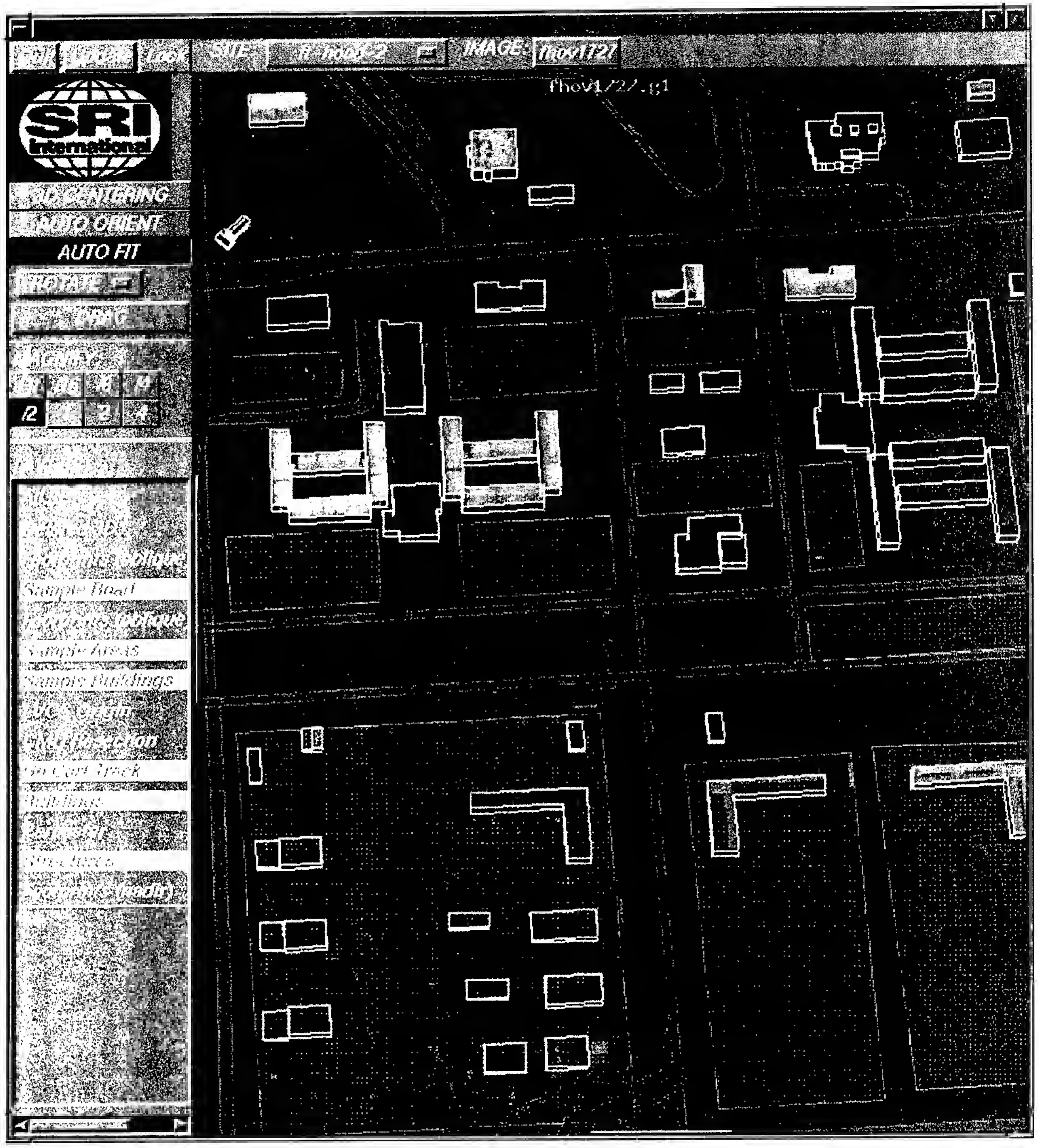


Figure 2: Part of a "ground truth" model being constructed of Ft. Hood.

lizing the WWW, this will allowing continuous evaluation of our progress and foster the formation and strengthing of the APGD community. We envision a central repository on the web with links to raw and processed datasets, interactively constructed "ground truth" models and models constructed through the use of the best available automated techniques currently incorporated in the BOS. Hyperlinks in the models themselves, will allow the user to retrieve performance statistics and intermediate results that are relevant to the model construction process.

4 Summary

The objective of this APGD IFD effort is to develop, evaluate, and demonstrate the enabling technology that will permit rapid, robust, automated population of geospatial databases – especially in support of "virtual worlds" and image-exploitation applications. This technology does not currently exist; therefore, our primary concern in the first two years of this effort is research/technology development. This first phase of the APGD/IFD effort is not an attempt to construct a testbed, or a workstation, or even a completely integrated system.

The central problem our work will address is that of reducing the role of (or completely removing) the human operator from the "low-level" processes of constructing geospatial models from images and stored data. We will develop the technology to achieve robust (predictable, reliable) system operation employing automated algorithms.

Our approach is centered on a unique architecture that includes the following major components: a set of automated "feature extraction managers" and a "context-based algorithm control system;" a sensor calibration and control system that will support cross-sensor analysis of NTM, SAR, IFSAR, IR, panchromatic, and hyperspectral imagery; a collection of algorithms for feature extraction and attribution primarily concerned with roads, rivers, lines of communication, buildings, aerial features (land cover classification, material identification, water bodies, etc.), terrain, and man-made movable objects; and a persistent object-oriented blackboard to store our continuously updatable world model.

A primary theme of almost every aspect of our work is that of evaluation. In addition to the critical role evaluation plays in our context-based algorithm control system, there are many other reasons why the development of an effective evaluation methodology is a desirable goal in itself. For example, the ability to track progress and the ability to engineer systems which can meet specified performance/robustness objectives. We believe that a successful effort to develop and exploit evaluative techniques is equal in importance to the improvement of the feature-extraction algorithms being evaluated.

References

[Heller and Quam, 1997] Aaron J. Heller and Lynn H. Quam. The RADIUS Common Development Environment. In Oscar Firschein and Tom Strat, editors, *RADIUS: Image Understanding for Imagery Intelligence*. Morgan Kaufmann, San Mateo (CA), 1997.

[Heller *et al.*, 1996] A. J. Heller, P. Fua, C. Connolly, and J. Sargent. The Site-Model Construction Component of the RADIUS Testbed System. In *DARPA Image Understanding Workshop*, pages 345–355, 1996.

[Strat and Fischler, 1991] T. M. Strat and M. A. Fischler. Context-Based Vision: Recognizing Objects Using Both 2D and 3D Imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, October 1991.

Automatic Acquisition of Hierarchical, Textured 3D Geometric Models of Urban Environments: Project Plan

Seth Teller*
MIT Computer Graphics Group
545 Technology Square NE43-208
Cambridge MA 02139
seth@graphics.lcs.mit.edu, http://www.graphics.lcs.mit.edu

Abstract

This paper presents an overview of a planned system for automatic population of geospatial databases which represent urban exteriors as textured geometric model data. The salient features of the system are that 1) it employs "pose-imagery:" high-quality, digital still images, each timestamped, and annotated with a reliable estimate of the acquiring camera's absolute position and attitude; 2) it exploits both dense and sparse image features by sifting evidence for interpixel correlation across every image; 3) it addresses the combinatorial aspects of large-scale 3D reconstruction from very large number of images; and 4) it has no "human in the loop."

This work involves both engineering and research challenges. The engineering challenges include rapid acquisition of 6-DOF pose-instrumented, high-resolution imagery, and its insertion into a hierarchical three-dimensional data structure. The research challenges include dense reconstruction techniques using thousands of images, and the incremental construction of a textured three-dimensional model from selected, spatially related image subsets.

Also presented are an evaluation plan for the project, as well as a description of the synthetic, hybrid, and actual datasets to be acquired and processed.

1 Introduction

We have proposed the construction of a prototype system whose objective is the fully automatic population of geospatial databases (APGD) for built-up areas. We are developing algorithms to integrate ground-based, low-flyby, and satellite imagery; robustly handle illumination changes and occluding foliage; and gracefully yield multi-resolution models in the presence of data with varying quality. The generated models will have a form suitable for visual simulation, collision detection, change detection, line of sight simulation, and other physically-based simulation operations. We hope also to support model amplification (given additional observations) and modification (to incorporate the effects of, e.g., demolition, renovation, or construction activity. Finally, when data is being gathered, the system should be able to provide useful directives as to where further imagery is needed.

The objective of the system is to fundamentally accelerate and improve the population process for complex geospatial databases, surmounting the scaling problem and human dependencies posed by existing semi-automatic systems, and providing a fast new model acquisition capability for simulation operations. An operational system would also have significant commercial applications: entertainment, embedding of commercial databases, tourism, education, etc.

2 Objectives

The principal goal of the system is to populate textured geospatial databases in a fully

*Funding for this research was provided in part by the National Science Foundation under award IRI-9501937, in part by Intel Corporation, and in part by the Advanced Research Projects Agency of the Department of the Defense under contract DACA76-97-K-0002.

automated fashion, and to dramatically reduce the cost and time required to populate such databases as compared to existing semi-automatic (i.e., human-assisted) systems. Using many thousands of close-range images, rather than a few, and annotating them with high-quality pose estimates in an absolute Earth coordinate system, make the "footprint" of this effort quite distinct from those of existing efforts [Niem, 1994] [Collins *et al.*, 1995] [Streilein, 1995]. We hope to learn much and develop a class of useful 3D reconstruction algorithms novel from both research and engineering standpoints.

The fundamental engineering strategy employed by our system is the use of recently available global-positioning systems (GPS) and PC-hosted, high-quality accelerometry which yield continuous estimates of camera position and orientation in an absolute coordinate system. The resulting "pose-camera" will ease some of the most significant hindrances to achieving automated 3D reconstruction [Faugeras, 1993]. We have developed several novel algorithms that process pose-imagery to: establish sparse (edge) and dense (region) correspondences; identify regions of empty space; and reconstruct globally consistent 3D models (with confidence bounds) from local, occluded observations. Absolute pose estimates are used to avoid combinatorial blowup while processing very large numbers (thousands) of images.

Another principal system objective is that the quality and articulation of the reconstructed models degrade gracefully with the precision and resolution of the acquired pose and image data. Thus, pose imagery arising from "fast" drive-bys or fly-bys will yield crude but consistent 3D models. Higher resolution, or more densely sampled, image data will enable reconstructed geometry and texture of correspondingly higher fidelity (model "amplification" or "intensification"). We plan to validate these algorithms by contrasting their output with the product of traditional (human-effected) site modeling techniques.

3 Project Plan

We are developing both an innovative device (a pose-camera) and innovative algorithms (for 3D data organization and reconstruction) for automated population of a geospatial database. The geospatial database represents the geometric and reflectance characteristics of built structures observed in the physical world, as well as significant trees and vegetation. All data are modeled by templates, that is, canonical parametrized objects fit to multiple observations. The generated 3D model will support line-of-sight computations, physically-based collision detection, and other simulation modes. Associated reflectance information increases realism during simulation by enabling reillumination by synthetic light sources.

We will pursue four specific subtasks in order to demonstrate system feasibility. First, we will design and build a pose-instrumented digital camera. Second, we will deploy the instrument in a known test environment to determine its performance. Third, we will deploy the pose-camera in a complex, unmodeled environment, and develop software algorithms for deriving 3D textured geometry and foliage representations from the acquired pose imagery. Fourth, we will assess the system utility and cost by manually modeling some portion of the automatically-acquired dataset, and comparing traditional techniques with our methods.

We distinguish between three types of pose-imagery data used for algorithm development and testing. Simulated data is synthetic imagery with arbitrary specified pose, generated with standard geometric modeling and computer graphics (rendering) algorithms. Hybrid data is imagery acquired by a manually operated camera mounted on a tripod, with pose estimates derived *a posteriori* via semi-automatic photogrammetry [Horn, 1986] or other human assistance. Actual pose imagery is that produced by the operational pose camera, with initial pose estimates given by instrumentation and refined by automated numerical optimization algorithms.

4 Challenges

Engineering challenges in this project include rapid acquisition of 6-DOF pose-instrumented, high-resolution imagery, and its insertion into a hierarchical three-dimensional data structure. Instrumentation packages for determining absolute position and orientation exist, but (as commercially available) in a largely unintegrated state. We are assembling integrated, PC-hosted instrumentation which will maintain pose estimates through a combination of global positioning, accelerometry, and odometry. This instrumentation will be mounted on a wheeled, human-propelled acquisition platform (Fig. 1), along with a high-resolution digital camera, on-board PC and digital tape storage, and a battery-based power source.

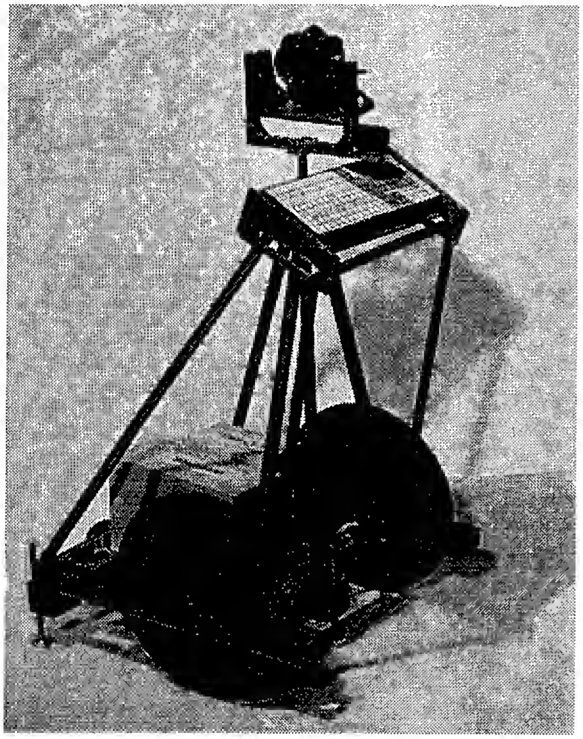


Figure 1: A model of the pose camera.

Raw high-resolution image data demands an enormous amount of storage space. An external-memory spatial data structure mediates storage and processing of both the annotated imagery and the reconstructed model data. Moreover, specular effects and changes in lighting conditions during acquisition will complicate data collection and matching efforts. However, these same factors also make possible estimation of the directional reflectance properties of each reconstructed surface.

Due to the size of the input and output datasets, only the most skeletal global operations, such as insertion of representations of acquiring cameras into a hierarchical spatial data structure, will be possible. Global, pairwise matching strategies are unworkable, since they would result in combinatorial explosion. Instead, local operations will correlate imagery acquired by proximal cameras, or suspected to contain observations of related physical structures. Incremental construction and insertion of reconstructed geometry from overlapping, adjoining imagery subsets must be supported. Associating spatially proximal cameras avoids both falsely negative and falsely positive matches that arise in schemes relying only upon temporal coherence (Fig. 2).

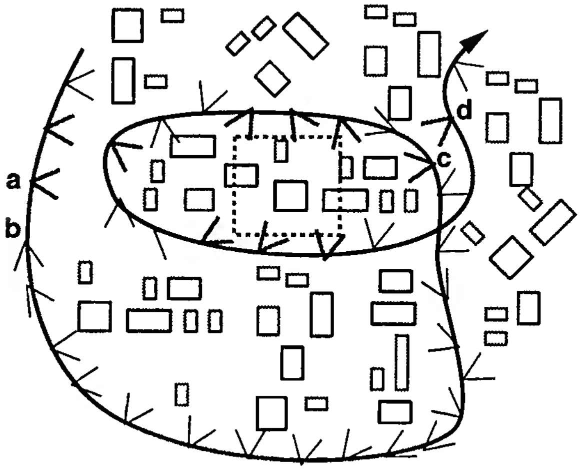


Figure 2: Temporal strategies can wrongly associate images (**a,b**) or fail to associate relevant images (**c,d**).

We augment traditional "sparse" (point- and edge-based) correspondence algorithms [Faugeras, 1993] with "dense" processing algorithms which correlate every pixel in every image with some other pixel set. The goal, of course, is to find that representation of the 3D world most consistent with the totality of 2D observations (pose images). Conservative error bounds will be maintained for all reconstructed features and structures. Confidence in this representation should increase with more numerous, more finely sampled, or more higher-resolution, acquired imagery.

Foliage imagery can be a significant hindrance to reconstruction of urban exteriors. We address this issue with a model-based foliage reconstruction scheme. Synthetic images of procedurally generated foliage are subjected to filters, and their responses associated with the foliage generator settings. Imagery of actually occurring foliage is then subjected to analogous filtering in order to identify similar procedural foliage. The objective is not to reconstruct foliage directly, but rather produce procedural foliage representations morphologically similar to those observed.

5 Evaluation Plan

We plan to evaluate the pose camera's accuracy by deploying it in a known (manually-surveyed) arena and testing its reported pose against independently derived estimates. We will establish mean and maximum errors in camera position and camera attitude as a function of time and distance from start, both for slow, regular motions, and for rapid translation and rotational acceleration.

We are completing a careful "ground-truth" survey of the Technology Square area. We will evaluate the quality of initial reconstructions by comparison to ground truth in the form of manual survey data, and architectural facilities data maintained by the campus Office of Facilities and Management. We will determine the accuracy and resolution of 3D feature reconstruction, and assess the faithfulness of the reconstructed datasets. We plan also to develop models of degradation of reconstructed data with loss of image resolution or pose accuracy.

During year two, we will attempt to achieve basic, block-based feature and building reconstruction of Technology Square and some portion of the MIT campus (from five to two hundred structures). The pose-camera will be manually moved through the relevant areas. Our initial system implementation will generate reconstructed geometry and texture data. Textures (reflectance maps) for each building facade will be generated by aggregating disparate pose images, to be compared to "ground truth" by reference to axis-aligned photographs of building facades acquired manually under measured, nearly diffuse lighting conditions.

Finally, we plan to evaluate the throughput and cost of the system according to several metrics. The rate of pose-imagery acquisition will be measured. The computational costs of 3D reconstruction will be assessed, along with dollar estimates of system overhead and per-feature cost. Finally, system operation will be compared, for a single large dataset (imagery of several hundred structures), with that of an expert human operating a traditional semi-automatic photogrammetry system.

6 Conclusion

We have described an ambitious, but feasible, vision of a system for fully automatic population of textured geospatial databases representing built-up areas, with no human in the loop except to direct motion of the pose-camera. The system design exploits recent advances in pose instrumentation. Geospatial data entities and associated textures are incrementally deduced from a large collection of images with pose estimates of varying accuracy. The system output is a collection of geometric entities, each with a conservative confidence bound, organized in a hierarchical spatial database suitable for external-memory algorithms such as those employed by real-time simulation systems. Because the acquired dataset is fully three-dimensional, it can be subjected to collision detection, line of sight, arbitrary lighting and atmospheric conditions, and other physically-based or phenomenologically-based simulation operations.

References

[Collins *et al.*, 1995] R. Collins, Y. Cheng, C. Jaynes, F. Stolle, X. Wang, A. Hanson, and E. Riseman. Site model acquisition and extension from aerial images. In *ICCV*, Cambridge, MA., 1995.

[Faugeras, 1993] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[Horn, 1986] Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[Niem, 1994] W. Niem. Robust and fast modeling of 3D natural objects from multiple views. In *SPIE Image and Video Processing II*, San Jose, CA., 1994.

[Streilein, 1995] A. Streilein. Integration of digital photogrammetry and CAAD: constraint-based modelling and semi-automatic measurement. In *CAAD Futures '95 Conference*, Singapore, 1995.

Knowledge-Based Automatic Feature Extraction

Ram Nevatia and Keith Price
Institute for Robotics and Intelligent Systems
University of Southern California
Powell Hall Room 204, MC-0273
Los Angeles, California 90089-0273
http://iris.usc.edu/Outlines/apgd-project.html

Abstract

Constructing geospatial databases is a tedious manual operation. Automatic 3-D feature extraction from 2-D images requires solving a number of problems. We present a plan to attack this task using a combination of tools: reconstruction and reasoning in 3-D, use of multiple sources of data, perceptual grouping, use of context and domain knowledge, use of previous maps and models, and limited use of human input where applicable.

1 Objectives

Geospatial databases are important for a number of battlefield awareness tasks such as mission planning, mission rehearsal, tactical training and damage assessment. Other applications include intelligence analysis for site monitoring and change detection. Geospatial database requirements may vary for the different tasks, but generally knowledge of terrain elevation, surface features and cultural features is needed. Our task focuses on the automatic detection and description of cultural features, particularly buildings.

The importance of cultural features for multiple tasks is quite clear. A mission plan in urban or semi-urban environments must consider buildings and similar structures. These may be the targets of an operation or assets to be utilized in the mission. Road networks and buildings are also key components in a simulated scene for such environments. Damage assessment reports also may be required for the infrastructure. For site monitoring, cultural features are not only of interest in themselves but also provide context for the detection of other features such as vehicles. The cultural features in a database also help in orienting an analyst to the observation of a new image by registering the image to a site model and by pointing out the major known features of interest. Thus, the construction of geospatial databases containing cultural features is of key importance in all aspects of battlefield environment, from initial site monitoring to mission planning and rehearsal, in the execution of the mission itself, and then in an analysis of the results after an action.

Some commercial *softcopy* systems for constructing geospatial databases are available. These can be helpful in the process of recording the data and carrying out many photogrammetric computations. However, the extraction of the important features, particularly the cultural features, largely remains a manual task. At best, these systems allow a user to choose parameters of a prototype shape; the large number of parameters can make this a tedious and inefficient process. There has been some progress on automated detection in research systems but the capabilities remain limited. We need to significantly expand the classes of objects that can be modeled, the conditions under which they may be imaged and to make the systems more robust and reliable.

2 Research Issues and echnical Approach

The problem of 3-D feature extraction is difficult in many ways. Low level segmentation techniques (such as line detection) give incomplete and imperfect results. Object boundaries may be incomplete and many extraneous boundaries, due to markings,

* This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by Topographic Engineering Center of the U.S. Army.

shadows, noise and other features, may be present as can be seen in the example of Figure 1. The system must complete the desired object boundaries and discard the extraneous ones. It must also infer the 3-D structure of the objects, which is not explicit in an intensity image and needs to be inferred.

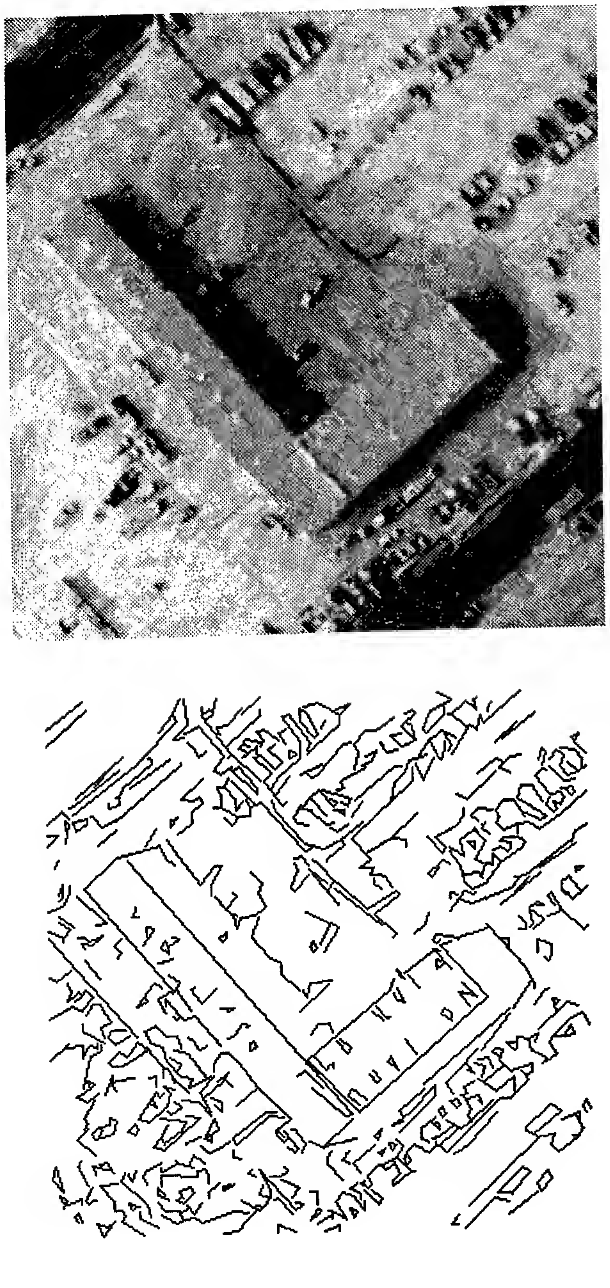


Figure 1 Image (top) and extracted line segments. (bottom)

The segmentation and 3-D description problems become easier if a 3-D range sensor, such as LADAR or IFSAR is available, as image discontinuities correspond more directly to object or surface boundaries. However, some extraneous boundaries due to noise and other sources would remain. Also, sensors such as IFSAR can not provide a complete range map, typically the data has areas of missing elements and may contain some points with grossly erroneous values.

In the absence of an ideal range sensor, we can infer depth from multiple images. However, this requires finding corresponding points or features in two or more images, which is a complex task in itself. Also, such a process gives sparse information about 3-D features which must still be grouped appropriately to extract the desired objects.

We plan to overcome the difficulties of 3-D object detection and description by using a combination of tools: reconstruction and reasoning in 3-D, use of multiple sources of data, perceptual grouping, use of context and domain knowledge, use of previous maps and models, and limited use of human input where applicable. This combination is shown schematically in Figure 2. Reasoning in 3-D has many advantages, as the objects we desire are 3-D objects rather than 2-D surface features. The use of multiple images and multiple sources of data will aid in the problems of 3-D reconstruction and also in resolving ambiguities that may be present in a single image or in images taken from a single sensor. Perceptual grouping is an essential step in selecting and organizing lower level features into meaningful objects. The use of context and domain knowledge will help with the reduction of ambiguities, with help in attribution and in some cases help with choosing the appropriate algorithm parameters. In cases where previous maps or models exist, these can be used to reduce the needed work; instead of building complete models, the system can focus on change detection and updating. Even with use of these multiple tools, some human interaction may be required, either to initiate the automatic processes correctly or to edit the results. We do this when necessary but minimize the effort required from the human operator.

We now describe the elements of our approach in more detail:

a) Reconstructing and Reasoning in 3-D
Explicit 3-D representations are needed for many applications. Knowledge of 3-D also makes the task of segmentation easier as we can find depth and ori-

entation discontinuities and separate them from image discontinuities caused by sources such as markings, shadows and highlights.

Some sensors, such as IFSAR, provide direct measurement of height, however, this information may be incomplete and may contain significant isolated errors. Height can also be extracted from multiple 2-D images by making correspondences between features visible in two or more images. The process of finding correspondences is, however, a difficult task in itself. One key issue is the level of features at which this matching should be performed. Low level features are easier to compute but more ambiguous. Higher level features are easier to match but difficult to infer reliably from image data. We use a hierarchical approach where features are matched at various levels and the matching at one level helps compute features at higher levels. We have constructed an early version of a hierarchical multi-image system [Noronha & Nevatia, 1997].

b) Use of Multiple Sources
Multiple sources of data can help in many ways. One is in estimating heights even if each sensor only gives a 2-D image. We can also take advantage of other characteristics of the sensors as the different sensors may be better at extracting different kinds of information. For example, IFSAR images have height information. In an ideal case, certain kinds of features, such as a flat horizontal roof, may be found just by thresholding on height. Surface boundaries may be found by first derivative discontinuities in the image. However, IFSAR data is likely to contain *holes* and some of the values may have gross errors. Thus, results obtained by simple processing are not likely to be of sufficient quality for accurate building models. We believe that IFSAR images will be highly useful for detection, but detailed description and delineation may require use of supplementary sensor data.

This is illustrated in Figure 3. The results using IFSAR alone are obtained by thresholding height to get roof locations and the EO results are derived by our monocular building detection system with the approximate location given by the IFSAR roof areas; these results are shown only to indicate the potential of combining sensor modalities and not to imply these problems are solved.

Multispectral and hyperspectral images, if available

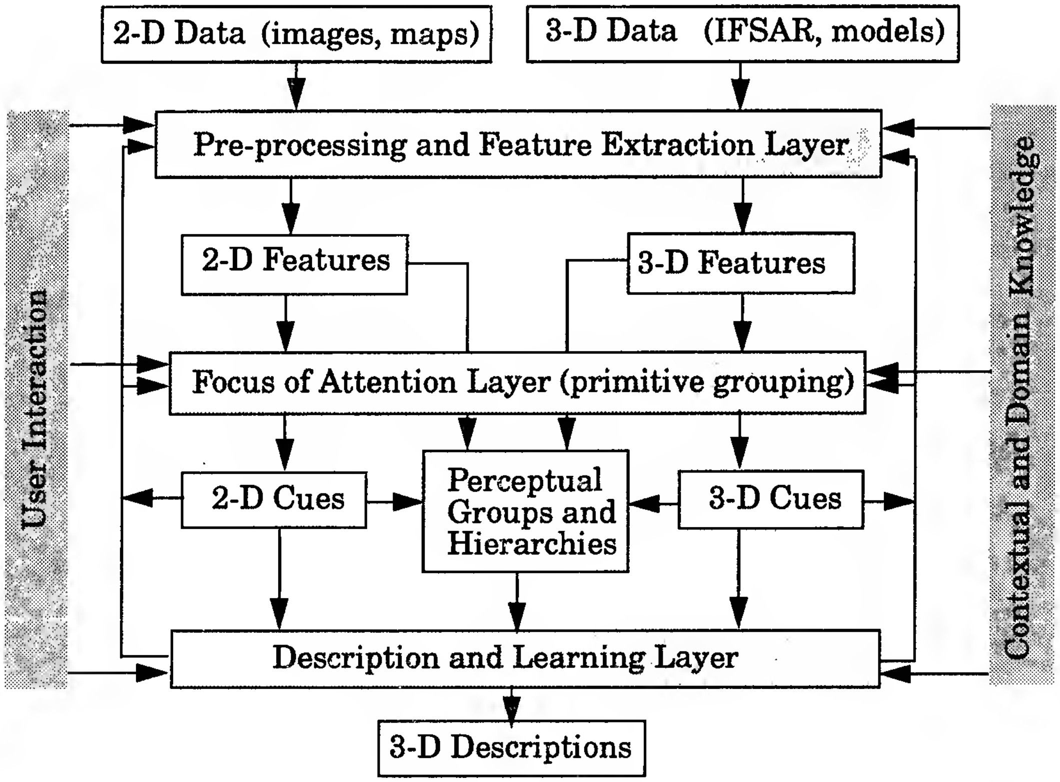
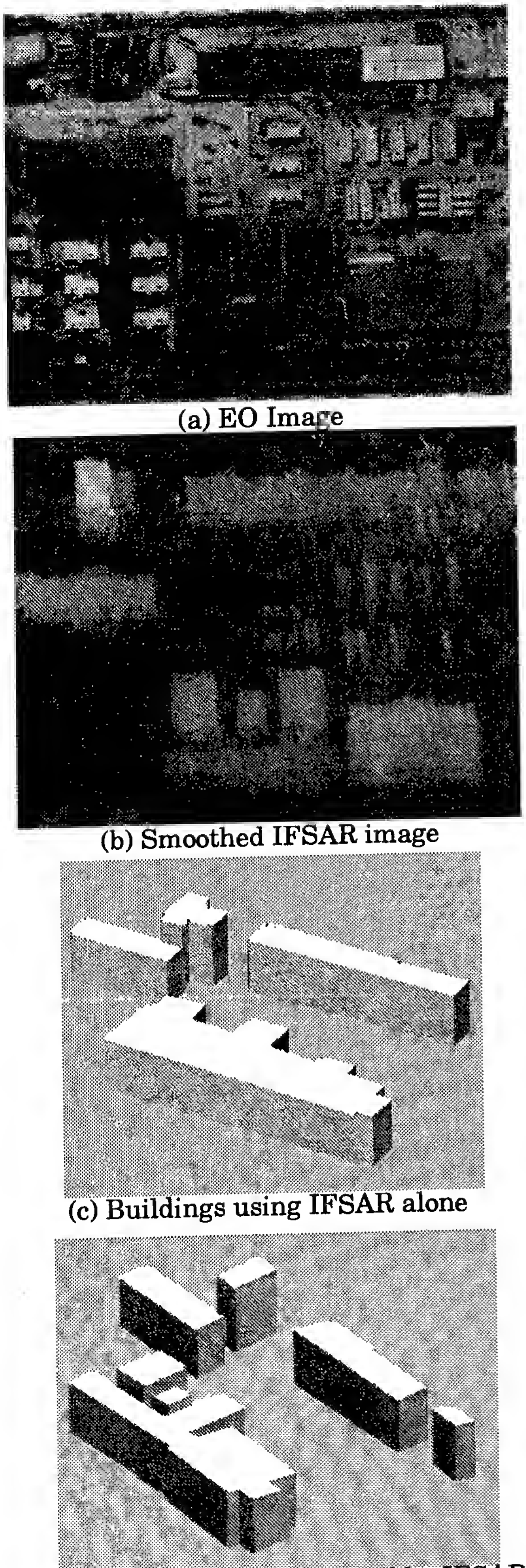

2-D Data (images, maps)
3-D Data (IFSAR, models)
User Interaction
Contextual and Domain Knowledge
Pre-processing and Feature Extraction Layer
2-D Features
3-D Features
Focus of Attention Layer (primitive grouping)
2-D Cues
Perceptual Groups and Hierarchies
3-D Cues
Description and Learning Layer
3-D Descriptions

Figure 2 Schematic Diagram of the 3-D Grouping Approach



(a) EO Image

(b) Smoothed IFSAR image

(c) Buildings using IFSAR alone

(d) Buildings using EO cued by IFSAR

Figure 3 Building extraction using EO and IFSAR

can aid in the processing in many important ways. They can be useful in detecting vegetation and surface material and may help in distinguishing between natural and cultural features. They can be useful for distinguishing shadow regions (shadow regions have different spectral properties, in a normal color image they typically have a higher blue component from scattered light). Hyperspectral images should also aid in segmentation of images as different kinds of materials can be clustered by spectral properties more easily than by intensity alone. IR images may also be useful for detecting vegetation and some components of a 3-D structure.

c) Perceptual Grouping

Much of the information about an object resides in its geometric properties. However, the geometry of the object needs to be inferred from the image. Early segmentation processes typically give low-level features that are fragmented and the desired object features are not separated from those in the background or those due to surface markings, shadows or noise. The goal of perceptual grouping is to organize these features into meaningful groups that give the desired object geometry. Use of certain kinds of sensors may provide better connected features with fewer distracting ones, but the process of perceptual organization is still required.

Our approach to perceptual grouping is a hierarchical one. Lower level features, such as lines, are grouped into successively higher levels, such as parallel or symmetric lines, which in turn are grouped into features that may correspond to surfaces of desired objects. The surfaces may then be further grouped to give volumetric objects. Multiple hypotheses are possible at each level. Our approach is to select among them only at levels where sufficient information is available to do so. This results in many hypotheses being generated at each level. Also, the grouping process may use either 2-D or 3-D features, but the final *verification* step for the objects should use 3-D reasoning wherever possible.

The properties that are used for grouping and for selecting among possible groups are of key importance. We believe that these properties should derive from an analysis of expected invariances for classes of objects under various imaging conditions. For example, we know that surfaces of a rectangular parallelepiped will project to parallelo-

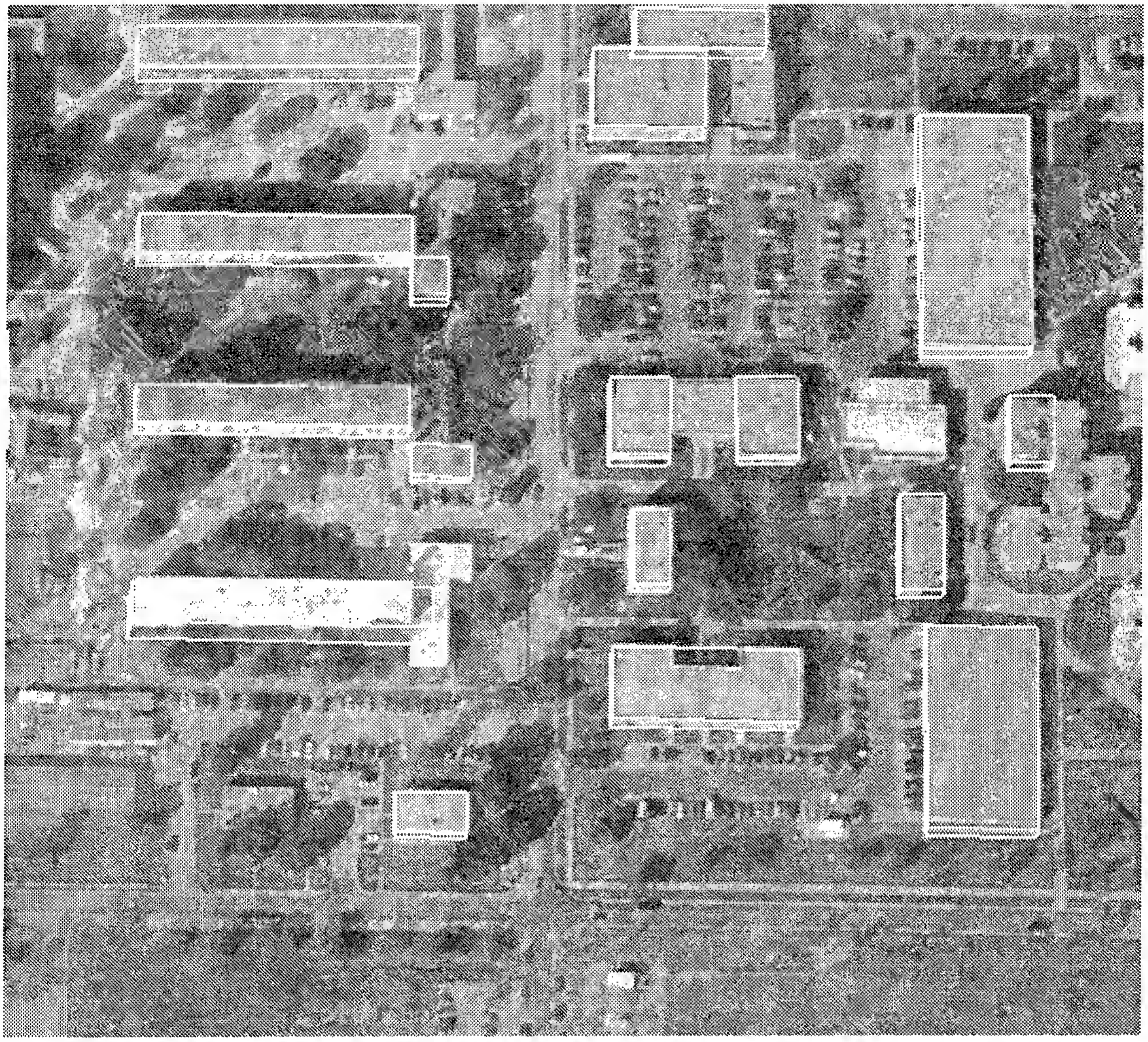


Figure 4 Buildings detected by multi-view hierarchical system.

grams under orthography (generally applicable when the sensor is relatively far compared to the size of the object). Next, we show an example of building detection using multiple images. We have developed a system that combines matching and grouping operations in a hierarchical fashion [Noronha & Nevatia, 1997]. Figure 4 shows the results of this procedure on a portion of the Ft. Hood dataset using three separate views. This is one of the most difficult parts of the Ft. Hood scene; trees obscure some of the sides and buildings have roofs at multiple heights. Note that there are no false alarms and most of the buildings are detected and delineated correctly.

In past work, it has been common to use geometrical properties. With the availability of multiple sources, grouping will need to use features from multiple images and combine them depending on the sensor characteristics. Combinations using geometric properties will be easier than combining sensor level data. We will also need a method to accumulate evidence of objects from a variety of uncertain sources.

d) Context and Domain Knowledge

Context and domain knowledge are important sources of information for object extraction. Presence of one set of objects can help reinforce or suggest the presence of others. For example, in an airport complex, the hypotheses for an airplane and a terminal reinforce each other if proper relations exist between them. Extraction of a certain road, or a transportation network in general, helps in identification of a certain building and in predicting locations of other buildings. Extraction of a road and river provides cues to the location of a bridge and so on.

Domain knowledge also helps us choose the tools that are appropriate for the task and in choosing parameters or rules for the algorithms. Different settings or methods may be appropriate for processing in rural or urban areas, or in areas with or without heavy vegetation, or in presence or absence of snow.

e) Use of Previous Models and Maps

In many cases of interest, previous maps and models may exist and the task may be one of updating or extending them. In this case, we need to register the new images with the existing model, find the differences between the two and update the models with descriptions of the new and changed features. The process of finding the differences consists of computing the expected visible features (for the current imaging environment) and verifying whether these features are present in the image. Regions of significant difference can invoke the model construction process. Some capabilities of image to model registration and model validation have been developed as part of our RADIUS effort [Huertas, *et al.*, 1995; Huertas & Nevatia, 1997]

f) Human Interaction

Even though the goal of this effort is complete automation, it is likely that the systems that can be developed in the near term will not be perfect and will miss some objects or find incorrect ones. A mechanism is needed to edit and correct them. This should not require a user to invoke completely manual procedures; in many cases, it is sufficient for the user to provide some hints to the automatic system to recompute and correct the problems. We have some experience with such an approach where in some cases a missed building can be found simply by the operator indicating the approximate location of the building and a possible cause of the failure [Heuel & Nevatia 1996]. Sometimes, more precise interaction may be needed, but it should still not be necessary to revert to a complete manual system. For example, if the size of the roof is corrected by the user, the height can be recomputed automatically using the same procedures as the automatic extraction system. This work is reported on in more detail in [Huertas & Nevatia 1997].

3 Evaluation Plan

We are developing relatively complete, end-to-end systems that start with images (and some collateral data when available) and produce 3-D object models. This makes it easier to establish evaluation metrics and to test the systems. We describe some metrics and an evaluation methodology below.

3.1 Metrics

The following metrics capture issues in evaluating extraction results:

1) **Detection rate:** How often are the desired features detected? This can be in terms of the absolute number of detected objects or may be by some weighting (such as by size or by importance). We consider a feature to be detected, if there is any overlap between a detected feature and a desired feature (this could be modified to include a certain amount of overlap or a certain minimum accuracy of the model).
2) **False Alarm rate:** This measures the frequency of mistaken detection. A feature is considered a false alarm if it does not overlap with any desired feature (of the detected class). Again, the rate may be measured in terms of number of objects or by some weighting.
3) **Accuracy of Models:** This is more difficult to measure. We can measure errors in 2-D or 3-D. Typically we want to know the accuracy in terms of size, shape and location. Size error can be computed in terms of volumes or by other parameters such as area and height. Shape differences may be harder to characterize, except perhaps by the amount of overlap (in 2-D or 3-D). The error metric can be made specific to a shape, for example, for a rectangular structure, measurements of the three sides and the center may suffice.
4) **Confidence Factor:** We expect that our systems will be able to assign confidence factors to the detected features (and even to components of these features if necessary). These could be included as modifications to the above measures. For example, a false alarm indicated with lower confidence could be counted as being less severe than one with a high confidence.

3.2 Testing

For evaluations to be meaningful, the system must be tested on a wide variety of images that contain a variety of desired objects in a variety of environments, and imaged under a variety of conditions (possibly with a variety of sensor types). The results need to be characterized as a function of these variables.

3.3 Demonstration Plan

Our demonstrations will consist of taking input images, and any available collateral data, and displaying the results of our APGD algorithms and to compare them, in some cases, with hand provided *ground-truth* results. We expect to demonstrate results on a variety of objects using a variety of imagery sources. The system will be designed to run autonomously but some human interaction, either to initiate the tasks, or to edit the results may be allowed.

We will also aid in integrating our system with the system to be developed by the APGD IFD contractor and demonstrate our systems in a larger context. We intend to develop our software using the RCDE environment which should simplify integration with the IFD contractor.

References

[Chung & Nevatia, 1992] C.-K. R. Chung and R. Nevatia, "Recovering LSHGCs and SHGCs from Stereo," In *Proceedings of the DARPA Image Understanding Workshop*, San Diego, CA, January 1992, pp. 401-407.

[Heuel & Nevatia, 1996] S. Heuel and R. Nevatia, "Including Interaction in an Automated Modeling System," in *Proceedings of Image Understanding Workshop*, Palm Springs, CA, February 1996, pp. 429-434.

[Huertas & Nevatia, 1996] A. Huertas and R. Nevatia, "Including Interaction in an Automated Modeling System," in *Proceedings of Image Understanding Workshop*, New Orleans, LA, May 1997.

[Huertas, *et al.*, 1990] A. Huertas, W. Cole, and R. Nevatia, "Detecting Runways in Complex Airport Scenes," *Computer Vision, Graphics, and Image Processing*, 51(2):107–145, August 1990.

[Huertas, *et al.*, 1995] A. Huertas, M. Bejanin and R. Nevatia. "Model Registration and Validation", in *Proceedings of the Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Ascona, Switzerland, Birkhaiser, March 1995, pp 33-44.

[Lin *et al.*, 1995] C. Lin, A. Huertas, and R. Nevatia, "Detection of Buildings from Monocular Images", in *Proceedings of the Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Ascona, Switzerland, pp 125-134, March 1995.

[Medioni & Nevatia, 1984] G. Medioni and R. Nevatia, "Matching Images Using Linear Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):675–685, November 1984.

[Medioni & Nevatia, 1985] G. Medioni and R. Nevatia, "Segment-Based Stereo Matching," *Computer Graphics and Image Processing*, 31(1):2–18, July 1985.

[Mohan & Nevatia, 1989a] R. Mohan and R. Nevatia, "Perceptual Organization for Segmentation and Description," in *Proceedings of the DARPA Image Understanding Workshop*, Palo Alto, California, May 1989. Morgan Kaufmann Publishers, Inc.

[Mohan & Nevatia, 1989b] R. Mohan and R. Nevatia, "Segmentation and Description Based on Perceptual Organization," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 333–341, San Diego, California, June 1989.

[Mohan & Nevatia, 1989c] R. Mohan and R. Nevatia, "Using Perceptual Organization to Extract 3-D Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1121–1139, November 1989.

[Noronha & Nevatia, 1997] S. Noronha and R. Nevatia, "Detection and Description of Buildings from Multiple Aerial Images," in *Proceedings of the DARPA Image Understanding Workshop*, New Orleans, LA, May 1997

Research in the Automated Analysis of Remotely Sensed Imagery: 1995–1996

David M. McKeown, Jr.
Michael Bowling
G. Edward Bulwinkle
Steven Douglas Cochran
Stephen J. Ford
Wilson A. Harvey
Dirk Kalp
Jeff McMahill
Chris McGlone
Michael F. Polis
Jefferey A. Shufelt
Daniel Yocum

Digital Mapping Laboratory
Department of Computer Science, Carnegie-Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213–3891

Abstract

This paper presents an overview of the program of research at the CMU Digital Mapping Laboratory in the analysis of remotely sensed imagery and the construction of virtual world databases. We report progress in the areas of digital photogrammetry, automatic and semi-automatic building extraction, road extraction, stereo analysis, multispectral/hyperspectral image analysis, and virtual world construction for distributed simulation.

1 Introduction

This paper presents our annual overview of research conducted at the Digital Mapping Laboratory (MAPSLab), Computer Science Department, Carnegie Mellon University. The MAPSLab is supported under two DARPA research programs. The Rapid Construction of Virtual Worlds (RCVW) program supported by DARPA/ISO/SE, USATEC, NIMA/TMPO, and DMSO has focused on the end-to-end construction of visual simulation databases. This ranges from cartographic feature extraction, intensification of standard product spatial spatial databases, selection, generalization, and integration

Our research under the Automatic Population of Geospatial Databases (APGD) program supported by DARPA/ISO/IU has a much narrower scope of effort. Here we are addressing research toward automating cartographic feature attribution using high spatial resolution hyperspectral imagery, in combination with our existing cartographic feature extraction (CFE) systems running on panchromatic imagery.

In this section we briefly overview the research activites conducted by the MAPSLab in each of these related research areas.

1.1 RCVW Research

The purpose of our RCVW contract is to conduct a program of basic research in image understanding and automated cartography to support the efficient representation and rapid construction of virtual world databases. Such a system would take available panchromatic and multispectral/hyperspectral imagery, existing map and digital elevation models, knowledge about particular environmental and geographic conditions and automatically produce a virtual world database tailored to a particular simulation requirement. While currently far beyond the state-of-the-art, the research that we proposed addresses several issues, which when solved, will bring us closer to the goal of low cost construction of virtual world databases in a timely manner, produced according to standards that allow for sharing and interoperability.

Our activities are focused on three related research areas: efficient representation of virtual world, spatial database intensification to provide improved timeliness and level-of-detail for virtual world database construction, and the analysis of multispectral imagery for surface material characterization. Our research in cartographic feature extraction to support geospatial database intensification for automated

The Digital Mapping Laboratory's WWW Home Page may be found at: http://www.cs.cmu.edu/~MAPSLab simulation database construction is described in Sections 2, 3, and 4.

Section 6 gives a brief overview of our work in virtual world database construction. Progress in efficent representation and cartographic data processing is described in Section 6.1 while examples of real-world database construction used for actual exercises are described in Section 6.2.

1.2 APGD Research

The purpose of our APGD research is to investigate the use of high-resolution hyperspectral imagery will enable more detailed and accurate *surface material attributions* for simulation databases, especially in complex urban areas. Fusion of spectral information and derived surface materials with existing building and road extraction systems will greatly improve both the performance of such systems, by enabling hypothesis verification based on material type, and the cartographic utility of their output, by the addition of semantic attributions such as material type.

The focus of most cartographic feature extraction activity is the production of geo-spatial geometric descriptions of manmade structures and of the nature terrain. Under the APGD project we will investigate the automated extraction of *semantic attribution information* for these features by the fusion of hyperspectral and panchromatic imagery. While our ongoing research in RCVW with panchromatic imagery has focused on the geometric aspects of cartographic feature extraction, the generation of detailed surface material maps as well as the attribution of the composition of man-made objects has not until now been the subject of detailed analysis. Section 5 describes the acquisition of the HYDICE data, previously funded under the RCVW program, and the ongoing research in highly accurate geometric registration and radiometic correction that we believe are required for high quality information fusion.

2 Automated and Semi-Automated Building Extraction

Building extraction from aerial images continues to be a major component of our research, and the ability to relieve or eliminate the manual burden of building model compilation becomes increasingly important in light of the need to populate simulation databases with *geospecific building models.* Our research philosophy for the building extraction problem has been, and continues to be, predicated on three factors: the use of rigorous photogrammetric camera models to model image and scene geometry, the assertion that multiple cooperative analysis techniques are necessary to achieve robust performance on a wide variety of complex urban and suburban scenes, and the thorough performance evaluation and analysis of our systems[McKeown, 1996; McKeown *et al.*, 1996c].

In previous work [McGlone and Shufelt, 1994], we showed that photogrammetric knowledge could serve a key role in increasing the performance of feature extraction systems. In Section 2.1, we discuss PIVOT, a new monocular building extraction system completed during the past year, which uses photogrammetric information at every phase of its processing to achieve increased performance on complex and dense scenes. In Section 2.2, we discuss improvements and modifications to SiteCity, a semi-automated multi-image building extraction system, and show examples of site models generated with this system. The results obtained from both the automated and semi-automated systems emphasize the importance of using rigorous camera models for feature extraction [McKeown and McGlone, 1993].

We have made substantial progress in performance evaluation and analysis of our systems. In Section 2.3, we describe the methodology for 2D image space and 3D object space performance evaluation of our automated building extraction systems on 83 test images covering 18 sites, as well as a more recent feature extraction experiment for the Rapid Construction of Virtual Worlds (RCVW) research program, which evaluated performance of our automated and semi-automated systems on 26 test images covering 4 sites over Fort Hood, Texas.

Our research in building extraction has followed the cooperative methods paradigm [Shufelt and McKeown, 1993], which states that no one technique will be sufficient to handle the variety of complex scenes that will be encountered in the real world. By developing systems with a variety of different error modalities and scene analysis techniques, and merging the results, it is possible to achieve robust performance on a wider set of imagery. Recent work in the fusion of monocular building models with dense elevation maps derived from stereo matching represents one application of the cooperative methods approach. Preliminary results of that work is described in Section 4.

2.1 Developments in automated building extraction

Traditionally, computer vision techniques for automated building extraction have neglected the use of photogrammetric camera modeling as a source of geometric information, instead treating the image as the sole source of information. This limited view of the problem has forced both region-based and feature-based techniques to make strict assumptions about image geometry and scene content; consequently, such methods exhibit poor performance on imagery where buildings are not easily segmented by intensity criteria alone, or where complex shapes are prevalent and oblique viewing angles violate assumptions about image acquisition geometry.

PIVOT (Perspective Interpretation of Vanishing points for Objects in Three dimensions), is a new fully automated monocular building system [Shufelt, 1996a; Shufelt, 1996c]. A major distinction between PIVOT and the systems preceding it is the thorough integration of photogrammetric modeling in all phases of the building extraction process. Another important distinction is the ability to model buildings as combinations of *primitives*, simple volumetric models of fixed shape but variable size [Biederman, 1985], which can be used to represent complex structures without the need for a large model library.

PIVOT has five major steps in its bottom-up processing flow:

1. *Vanishing point detection* is performed after preprocessing phases of edge detection and recursive line fitting. This process makes use of the camera model to locate dominant horizontal and "peak" roof orientations in the image features, and has been described in detail elsewhere [Shufelt, 1996b]. The result of this process is a set of line segments labeled with plausible 3D object space orientations.

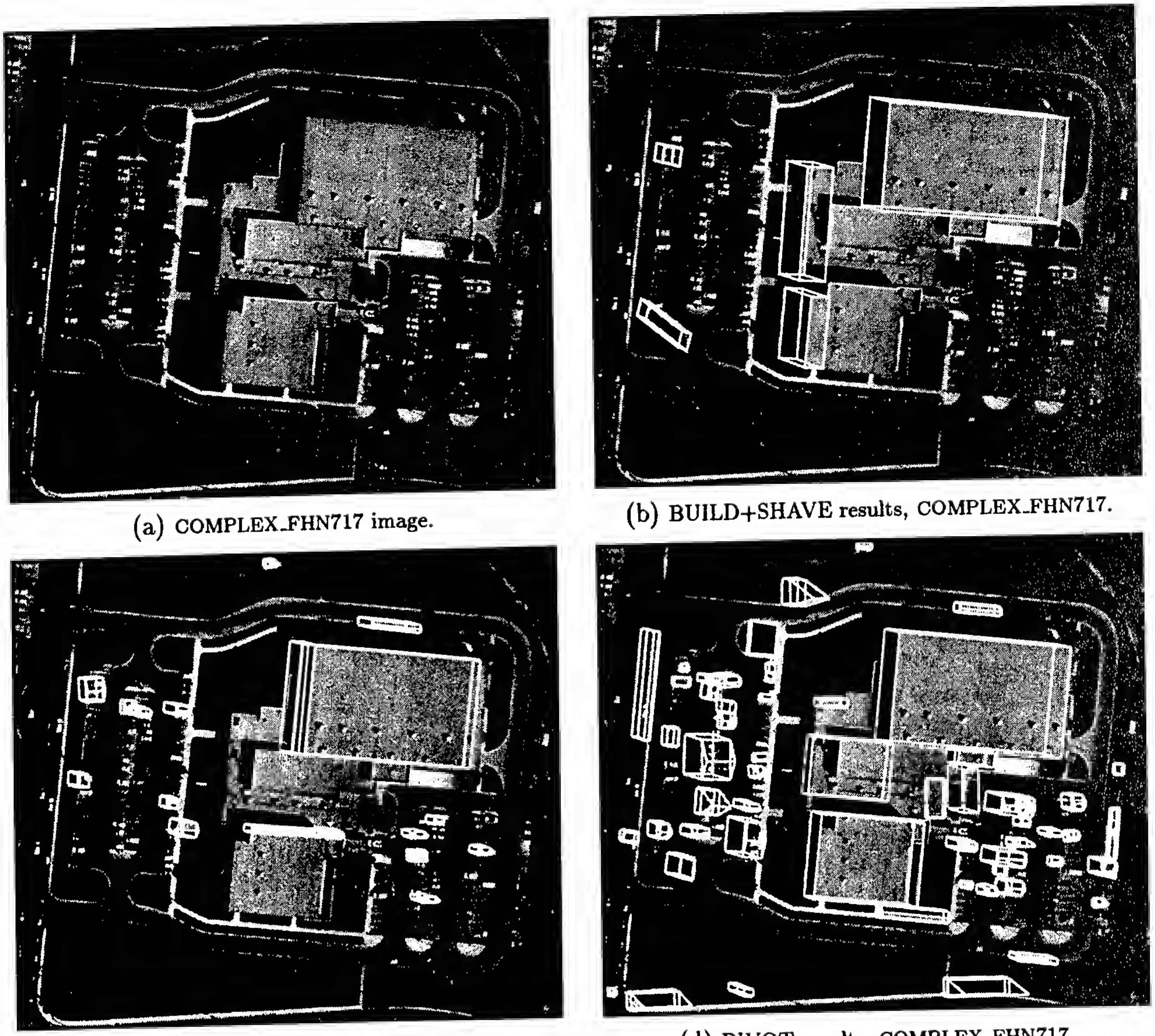
2. *Intermediate feature generation* uses the orientation-labeled line segments to construct corners and 2–corners (pairs of corners which share a segment). Each feature is subjected to a geometric consistency test which compares each possible orientation labeling with the orientation labels of the primitives. Only those features which have at least one legal labeling are accepted for further analysis. This process is also described in detail elsewhere [Shufelt, 1997].

3. *Primitive generation* uses the orientation-labeled intermediate features to perform a local search for plausible instances of primitives. These primitive instances are then subjected to evaluation for image support, and the best scoring primitive for each 2–corner is retained for further analysis.

4. *Primitive combination and extension* procedures are responsible for joining primitives together to form more complex objects, and for using primitives as starting points for searches for other primitives. One of the key ideas behind the use of primitives for 3D object modeling is that they represent a natural vocabulary for expressing complex shapes, eliminating the need for exhaustive representation of each unique shape that occurs in a scene.

5. *Building model verification* employs several tests to evaluate the final set of combined and extended primitives. The evaluation considers shadow support, photometric consistency across adjacent surfaces, edge support in the image, surface intensity homogeneity, and object space dimensions and geometry. It is worth noting that the shadow modeling is done in object space, unlike many other systems which use image space approximations. PIVOT also makes use of a new illumination constraint based on the known solar azimuth and elevation parameters to evaluate the consistency of surface intensity changes. The result of building model verification is a set of 3D wireframe building models in geodetic coordinates.

We briefly show the results from a PIVOT run, contrasted with the results of other building extractors developed at the Digital Mapping Laboratory in earlier research efforts. Figure 1a shows an image of a complex multilevel structure from Fort Hood, Texas. Figures 1b—1d show the automated building extraction results from BUILD+SHAVE, VHBUILD, and PIVOT, respectively. In this case, PIVOT



(a) COMPLEX_FHN717 image.

(b) BUILD+SHAVE results, COMPLEX_FHN717.

(c) VHBUILD results, COMPLEX_FHN717.

(d) PIVOT results, COMPLEX_FHN717.

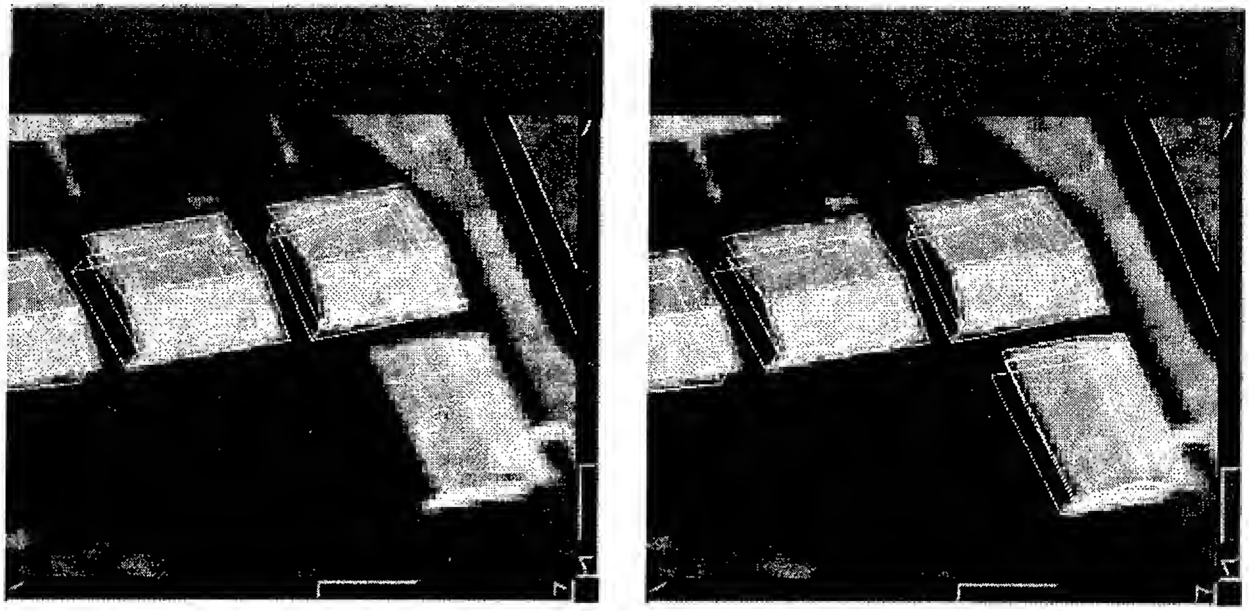
Figure 1: Automated building extraction results for a Fort Hood test scene.

is able to take advantage of its use of primitives in conjunction with a rigorous camera model to better delineate the 3D structure of the complex building, albeit with a higher rate of false positives. This contrasts with VHBUILD, which does not perform full object space modeling of shadows for verification, and with BUILD+SHAVE, which does not use camera geometry for hypothesis generation or verification.

While PIVOT can achieve impressive performance on several test scenes with no manual intervention, it is still the case that typical results from fully automated systems require manual editing before they are suitable for inclusion in a simulation database. The following section discusses recent progress in SiteCity, our semi-automated site modeling system.

2.2 Developments in semi-automated building extraction

SiteCity is a semi-automated multi-image site modeling system developed at the Digital Mapping Laboratory. SiteCity combines interactive measurement tools, image understanding techniques, and a rigorous constrained least-squares photogrammetric bundle adjustment package [McGlone, 1995] for site model generation. SiteCity models several generic classes of buildings with semi-automated tools, known as *automated processes*, which use IU techniques to alleviate the burden of manual measurement



(a) Overhang peaked roof buildings.

(b) Creating a rotated copy.

Figure 2: An example of SiteCity's *automated copy with rotate* process.

of structures in multiple images. Thorough descriptions of this modeling system and detailed usability evaluations have appeared elsewhere [Hsieh, 1996a; Hsieh, 1996b]; in this section, we focus on recent developments and additions to SiteCity.

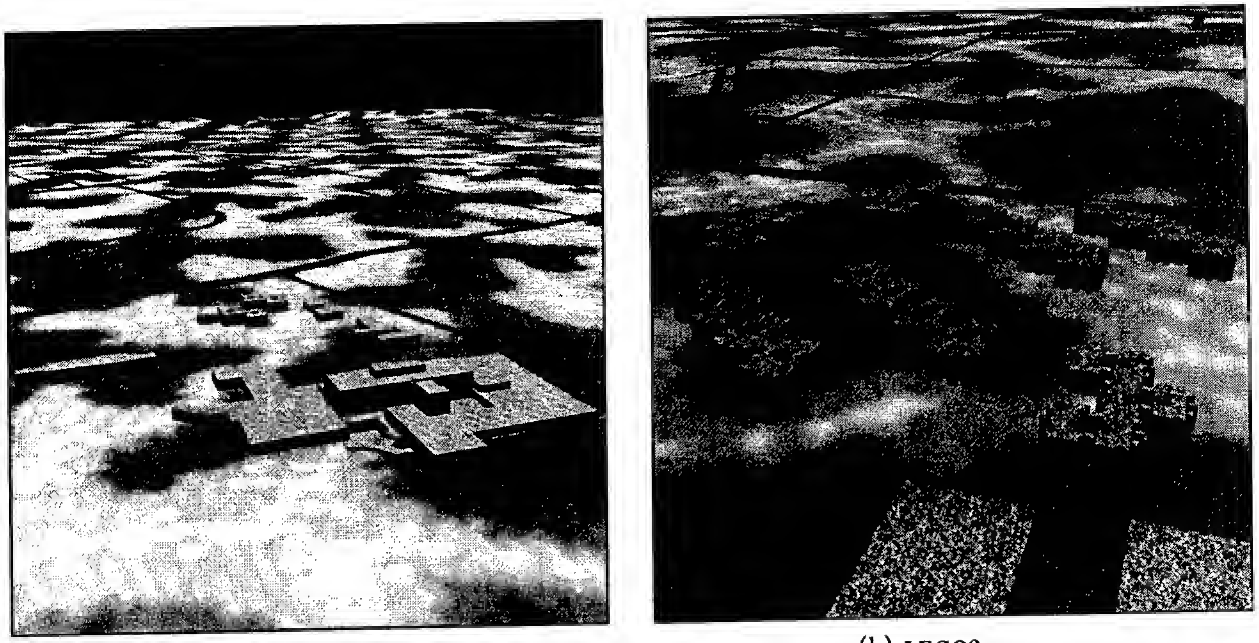
2.2.1 Model generation capabilities

SiteCity provides several tools for creating new building models with IU assistance as well as tools for constraining points, lines, and surfaces together to obtain models with correct geometry during the multi-image triangulation process. Recently, additions were made to each of these toolsets to improve the ability to measure and refine building models.

The original version of SiteCity contained a set of object constraint tools, which allowed a user to specify coplanarity and collinearity constraints among point sets, lines, and surfaces of building models. In addition to aligning building models, these tools were also useful for stacking buildings to form complex multi-level structures as well as joining walls of adjacent structures. SiteCity now has new constraint tools for coplanarity and collinearity, which allow the planes and lines to be constrained to horizontal and vertical orientations. These tools are useful for ensuring that constrained objects are not only tied together, but also lie in specific object space orientations. Work is underway to implement faster constraint solutions and more flexible constraint specifications.

SiteCity provided a number of automated processes, tools which could be invoked by the user to carry out certain measurement tasks automatically. One of these tasks, *automated copy*, provides a simple mechanism for repeatedly instantiating specific models at particular points, along a line, or within a polygon in the image; after the point, line, or polygon has been specified, SiteCity automatically finds instances of the model in the vicinity of these *copy regions* and performs multi-image matching. However, the copy tool only performed rigid translations of the model to be copied, which meant that rotated versions of the same model had to be measured separately. A new addition to SiteCity is the *automated copy with rotate* tool, which allows the user to specify a rotation angle in the xy-plane with a copy region. SiteCity then looks for instances of the model in the copy region with the desired rotation angle.

Figure 2a shows several peaked roof buildings with overhanging roofs, of which all but one share the same orientation. The remaining building is identical to the others except for a rotation of 90 degrees, and in previous versions of SiteCity, measurement of this rotated building would have had to start from scratch. Automated copy with rotate now allows a user to simply specify a rotation angle and click on



(a) STEW scene.

(b) LEGO2 scene.

Figure 3: Views of SiteCity models in Fort Hood simulation database.

the building, and SiteCity performs the rotation and does matching to position the rotated copy of the building. The result of this process is depicted in Figure 2b. This feature extends the range of situations in which SiteCity can be used to quickly generate new ground truth sites.

2.2.2 User interface issues

Each building model measured in SiteCity is represented as a hierarchical 3D wireframe object. A building model can be queried at the volume, surface, line, or point level, as appropriate. Measurements can also be adjusted at different levels in the hierarchy, which gives added flexibility in positioning model points and boundaries. However, the original SiteCity interface required the user to select among the volume, surface, line, and point levels before any queries or adjustments could be done. SiteCity now incorporates a 2D selection mechanism which allows objects to be selected in point-and-click fashion, without the need to explicitly specify the desired type of geometric object to be accessed.

While SiteCity has been used to generate the ground-truth models used for simulation database population as well as automated building extraction evaluation, many of the IU algorithms incorporated in the system remain active topics of research. For example, the multi-image matching routine which SiteCity uses to position a model from one image in the other images continues to undergo experimentation. To provide better support for these experiments, SiteCity now provides a display tool to show which edge segments and corners were used by SiteCity to choose a particular building match. These display diagnostics are useful in determining where particular matching techniques succeed or fail, and how they might be improved.

2.2.3 Ground truth model construction

As noted earlier, SiteCity has been used to construct a wide variety of complex building models for use in evaluating the monocular building extraction systems, as well as populating simulation databases. To support experiments under the APGD and RCVW research initiatives, SiteCity has been used to generate ground truth building models for several test areas in aerial imagery covering Fort Hood, Texas. The site models for these areas are representative of the modeling capabilities of SiteCity.

Figure 3a shows a view of the STEW test scene, which is dominated by a large multi-level complex structure in the center of the scene. Note that the structure is not strictly rectilinear; the left side of the structure has an angled wing.

Also, the central receiving area of the building has a curved overhanging parking roof with a hole where the roof joins the main body of the building.

Figure 3b shows a view of the LEGO2 test scene, which consists of several clusters of rectilinear buildings with complex perimeters and multiple holes. SiteCity models these structures by constraining together multiple instances of the "lego" block shape, which is only measured once and then copied as necessary. This approach allows for fast semi-automated measurement of these structures, but also places heavy demands on the constraint solution package. We are currently working to optimize the constrained bundle adjustment solution, a very necessary step given the size of the problems addressed. For instance, the LEGO2 site has seven sets of buildings, measured on seven images. Each set of buildings was solved separately; four of the sets had 2462 parameters (448 points and 717 constraints), while three had 1874 parameters (336 points and 542 constraints). Fortunately, the predictable, sparse structure of the normal equation matrix lends itself to efficient formation, reduction, and solution techniques. We are also re-implementing constraints and building models in more efficient ways, and working on ways to generate better approximations for the parameters so that fewer iterations are required.

2.3 Performance evaluation and analysis

There has been a relative lack of useful performance evaluation in previous research in object detection and delineation in aerial imagery. Frequently, the output of building extraction systems is only presented graphically, without recourse to quantitative evaluation metrics, and in cases where the scene models are compared with ground truth models, the metrics used give a biased or unclear measure of system performance. Although some work in the literature has been accompanied by thorough and rigorous performance evaluation, this has been the exception rather than the rule.

One of the main thrusts of our research in building extraction, and our broader efforts in cartographic feature extraction, is the extensive application of rigorous performance evaluation and analysis of our research systems. We believe that detailed performance analysis using a wide range of unbiased metrics is crucial to understanding system behavior, and we have developed several such metrics during the course of our work in building extraction.

In our most comprehensive performance evaluation of building extraction techniques to date, PIVOT was compared to three of our existing monocular building extraction systems, BUILD, BUILD+SHAVE, and VHBUILD on 83 test images covering 18 sites, with widely varying scene content. The results of that study showed that each system achieves its best performance in different areas of the scene complexity space, supporting the cooperative methods approach for feature extraction. In particular, PIVOT was shown to have improved performance in scenes with high object density, high object complexity, and high image obliquity, consistent with the observations that PIVOT uses a richer representation of building structure and employs a rigorous photogrammetric camera model. A detailed description of this evaluation can be found elsewhere [Shufelt, 1996c].

In more recent work, we designed and disseminated a feature extraction study guide as part of the RCVW research initiative, for presentation at the Terrain Week '97 meetings. We evaluated SiteCity and our four automated systems on the four new sites (26 test images) described in that document. This test document, and the briefing charts describing our results for this study, are available on our website (`http://www.cs.cmu.edu/~MAPSLab`). The systems were evaluated along several dimensions:

- Time required to extract features, broken down by feature extraction processing phases and manual/automated phases
- Space requirements for feature extraction algorithms
- Level of geometric, topologic, and attribute information extracted
- 2D building extraction percentage, branching factor, and quality percentage, expressed in terms of true/false positive/negative classification of image (pixel) space
- 3D building extraction percentage, branching factor, and quality percentage, expressed in terms of true/false positive/negative classification of object (voxel) space

- Performance as a function of image obliquity (nadir vs. low oblique vs. high oblique)
- Performance as a function of building complexity (image space and object space)
- Performance as a function of building density (image space and object space)

As in the previous study, we again observed that different methods performed well in various parts of the scene complexity space. For many of the test images used in the Terrain Week report, we found that BUILD+SHAVE achieved the best results, due to the prevalence of nadir views. VHBUILD and PIVOT typically achieve better performance in oblique views, when more 3D building structure is explicitly available in low-level edge information.

These studies, and other related performance evaluation tasks we have undertaken in the past [Shufelt and McKeown, 1993; McGlone and Shufelt, 1994], serve as examples of thorough analysis to guide research into feature extraction techniques, supported by intensive compilation of ground truth models. Figure 4 depicts a portion of a simulation database covering Fort Hood, with over 240km of road data compiled semi-automatically by techniques described in Section 3, and 14 building test sites containing 241 buildings compiled with SiteCity, ranging from simple flat rectilinear structures to multi-level complex buildings with internal holes. While the compilation of such ground truth databases remains a time consuming process, we feel that ground truth construction constitutes an essential part of cartographic feature extraction research.

3 Road Network Extraction

Our previous work in automated road network extraction has explored the use of cooperative methods for robust feature extraction from high resolution aerial imagery ([McKeown and Denlinger, 1988; Zlotnick and Carnine, 1993]). We continue to build on this work, investigating the use of scalable, object-space methods for automated and semi-automated road network extraction. The problem of detecting and delineating road features is difficult because roads are more than linear features. At spatial resolutions with ground sample distances of 3 meters or better, roads are complex, man-made structures composed of lanes (and possibly lane markings), medians, shoulders, sidewalks, over and under passes, width/lane changes, etc. Road features can be partially or completely obstructed by vehicles, buildings, and/or overhanging vegetation. Extraction techniques should take advantage of the road structure available at high resolution, but they must also be scalable if they are to be tractable for use in constructing large-scale feature databases. The use of object-space based methods provides opportunities to use real-world knowledge about road design structure, as well as factor in knowledge about the parameters of data acquisition (*e.g.*, camera angle, sun angle, time of day, etc.)

In this Section, we present some of our recent work in automated and semi-automated road network extraction. In Sections 3.1 and 3.2 we describe recent improvements to road feature representation, the interactive extraction system, and discuss its performance evaluation. In Section 3.3 we present a new model, with preliminary results, for object-space road tracking.

3.1 Interactive road extraction

We are continuing to develop our research in semi-automated methods for road feature extraction. Our interactive road network extraction system, called Idl_Woof, uses our automated road extraction system to augment the manual extraction of road networks from aerial images. The user interface is built on our X11-based image display library (IDL). As briefly described in [McKeown *et al.*, 1996a], Idl_Woof allows the user to direct execution of the automated road finding and tracking processes, as well as permitting manual delineation and editing of road segments. We use Idl_Woof both as a test bed for experimental automated methods, and as a modeling tool for manually producing accurate road network ground truth.

Our recent work has focused on improvements to our underlying feature representations. The 2D feature representation and output formats have been extended to support 3D objects. Intersection models are now supported, and the user-interface has been augmented to support the creation of these models. The representation contains hooks for other feature models, such as parking-lots, bridges and over-passes, and assymetric roads. We have also added support for feature attributions, so that road types,

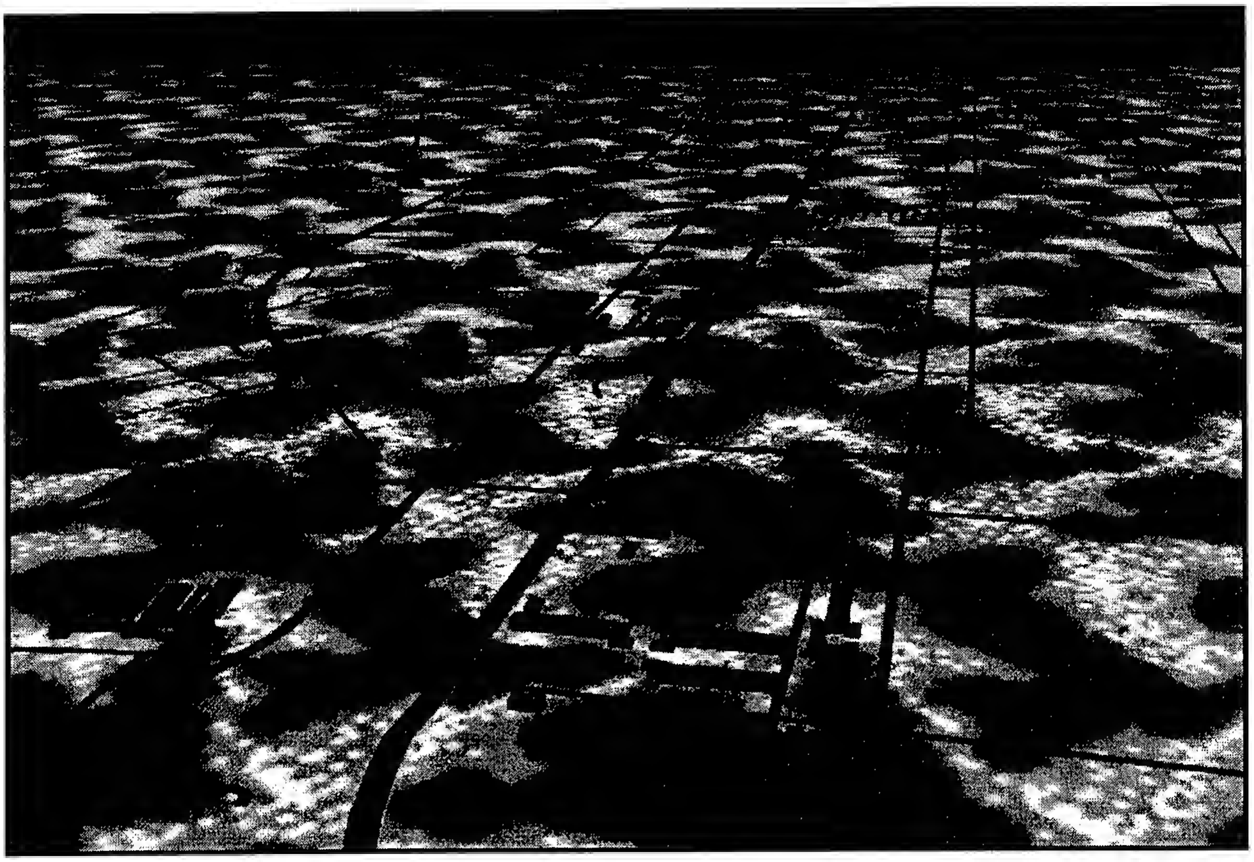


Figure 4: Fort Hood simulation database.

names, etc., can be carried along with the feature models. Annotated, 3D road network data from Idl_Woof was used to generate the scenes from the visual simulation database shown in Figure 4. Future work will concentrate on integrating these new feature models into the existing user interface.

Using Idl_Woof to compile road network ground truth is useful for several reasons. Extensive use of the interface helps us to identify and improve awkward or inefficient user interactions, as well as uncovering bugs in the representations. The generated ground truth data is used in the performance analysis of our automatic road network extraction system. Figure 5 shows an orthographic view of road and building ground truth databases compiled over the motor pool area of Fort Hood, Texas. Over 240km of road data was compiled from five separate 2.5km square images (approximately 7600 × 7700 pixels). The road networks from each image were joined (in object space), then merged with the building data, and finally overlayed on an orthophoto mosaic of the five source images. This result shows some of the detail and the extent of the road models one is able to produce with Idl_Woof. It also demonstrates the power of using rigorous photogrammetric and object-space representations.

3.2 Performance evaluation and analysis

The quantitative evaluation of a system's performance has remained a central theme in our feature extraction research. Though qualitative analysis is useful, it is fraught with subjective biases and, most times, overlooks the nuances that are often present in automated results. Using quantitative measures for evaluation permits a reliable determination of measurable gains/failures, thus determining the real utility of system modifications.

The basic method is to compare results against a manually compiled, high quality ground truth. This ground truth can come from a system such as Idl_Woof, or it could be an existing cartographic product, such as DTOP data. Comparisons against this ground truth should generate measurements that reflect the quality of the output, and lend some insight into where system improvements are most needed.

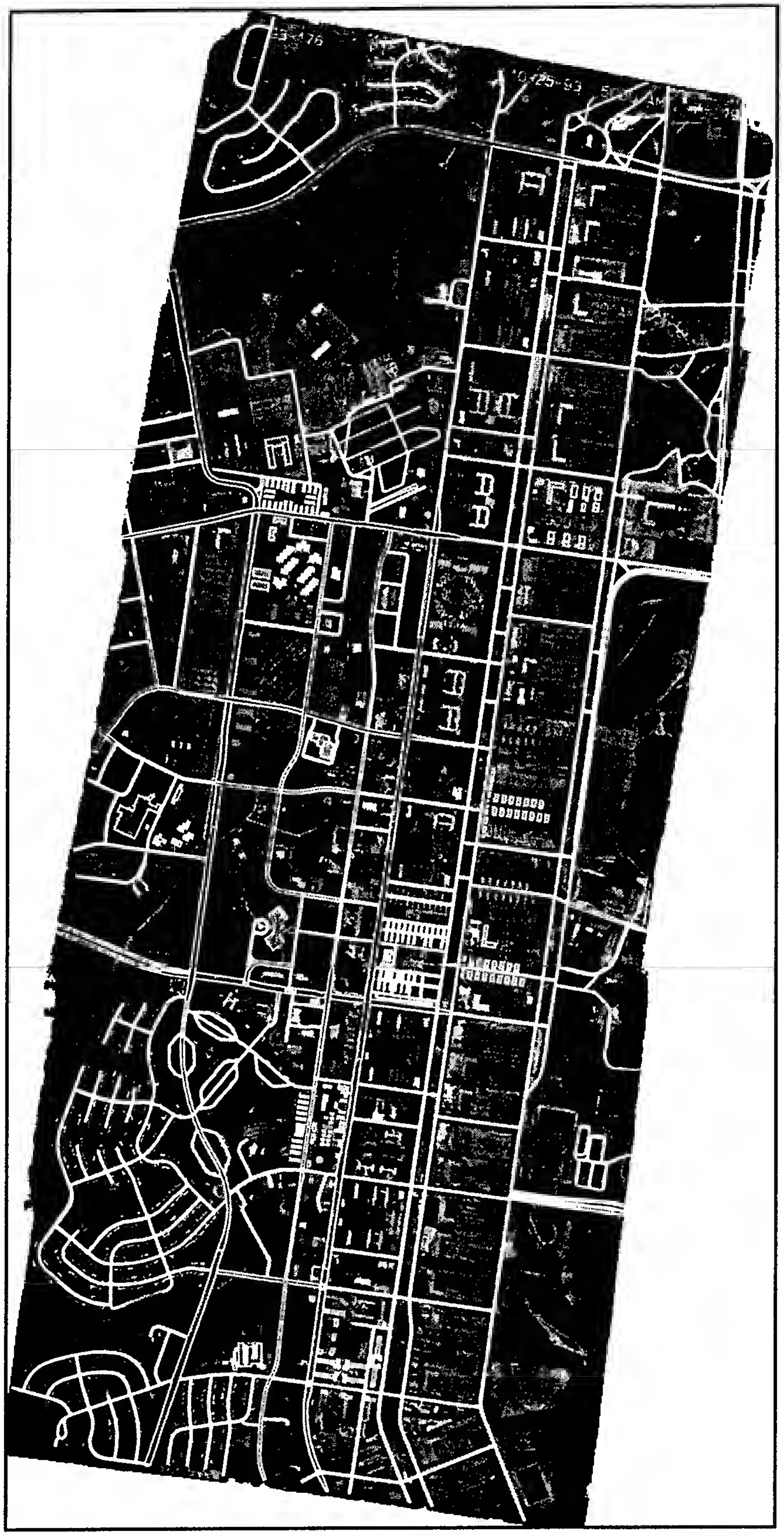


Figure 5: An orthographic view of road and building ground truth databases compiled over the motor pool area of Fort Hood, Texas. The feature data is overlayed on an ortho-photo mosaic of five nadir source images.

We used our existing automated road tracking system to generate results that we compared against our ground truth, shown previously in Figure 5. The results were generated by manually selecting a set of road starting points, then providing those points as input to our automated road tracker. Using manual starting points decouples the errors introduced by automated road finding from those introduced during tracking, permitting evaluation of road delineation independent of detection errors.

Table 1 presents the performance results of this comparison. The two data sets are compared in 2D to determine overlaps. For pixel-based measures (like redundancy), we create masks from both the ground truth data and the automated results and then compare the overlapping regions. For feature-base measures, we create a mask from the ground truth data, then compare individual road features against that mask. If a feature overlaps the ground truth mask by more than a given threshold (typically set to 75%), then that feature is considered a correct hypothesis. As with the building extraction performance evaluation, a confusion matrix is compiled, consisting of true-positives (TP), true-negatives (TN), false-positives (FP), and false-negatives (FN). From these numbers, several performance measures are computed:

- Branching factor (FP/TP) — Measures the number of competing hypotheses.
- Quality factor (TP/(TP+FP+FN)) — Measures the overall quality of the result.
- % GT Explained — The percentage amount of ground-truth that is covered by the result.
- % Current Hypotheses — The percentage of generated hypotheses that are considered correct.
- % Redundancy — The percentage of the generated output that overlaps itself. This is useful for determining how much extra work is being done.

It is important to use several performance measures in concert to obtain an accurate evaluation. For example, one could implement a classification method that labels all pixels in the input as roads, thus yielding a % GT explained of 100%. However, the branching factor would be extremely high, indicating that there is problem with the optimality of the classifier.

One can see from Table 1 that the number of correct road hypotheses is consistently high (70–80%), though the amount of ground truth covered is relatively low (20–30%). This is because the system fails to track about 70% of the input starting points, though the points that it tracks are accurately tracked. The branching factor measure is not as interesting here, as all the road starting points are considered correct since they were manually chosen. However, it is worth noting that the branching factor remains low, implying that the number of mistracks and overshoots are few. Future work will extend our performance measures to include performance analysis in 3D.

3.3 Toward a model for automated object-space road extraction

Recently, we have focused on improving the performance of our existing tracking system. As seen in some of the performance numbers presented here, it implements a conservative approach to extending road starting points. From an implementation standpoint, it is difficult to use this system to explore new tracker control scenarios. Additionally, there are many internal parameters that are pixel based, and it is difficult to change parameter settings based on "first principles". For these reasons and others, we were motivated to develop a new tracker design that would resolve most or all these problems.

Using our experiences with the current tracking system, coupled with our ideas about tracking scenarios, we have developed several principles that we've used to guide our new design. Trackers should be composable objects with well-defined interfaces so that new tracker control structures can easily be implemented. If trackers wish to share information, they must do so through a parent tracker, and that tracker must access and provide information only through the available interfaces. Coordinate systems are explicit, so that generating object-space features is straight-forward. Finally, the implementation of the model should be efficient, so that it is scalable, and so that it remains useful in an interactive context.

We represent a tracker as a generic object within the system. Each type of tracker (*e.g.*, cooperative tracker, surface tracker, edge tracker, verification tracker) is implemented as a specific instantiation of the generic tracker object. Trackers are not manipulated directly, but are created and modified as states in a finite state machine. Each tracker operation takes a state as input and returns a new state as output. This is a modular interface, and permits us to easily extrapolate forward or move backward through a tracker's history.

The model and operations were designed with specific applications in mind. A list of possible tracking scenarios includes:

- Cooperative tracker — A composition of two or more trackers. There are two versions of this:
  - Synchronous — Subordinate trackers advance in lock-step.
  - Asynchronous — Subordinate trackers are permitted to advance independently.
- Multi-Image tracker (sequential) — This tracker tracks in a single image and, when necessary, switches images in an attempt to continue tracking.
- Multi-Image tracker (parallel) — This tracker attempts to extract the same feature from 2 or more images simultaneously (a stereo tracker).
- Directed (guided) tracker - Assumes that "hints" have been given about where to find the road, such as a rough sketch or some preexisting data. It uses this information to extract a more detailed/accurate road feature.
- Multi-Resolution tracker — A hierarchy of corresponded data has been provided that will be used to extract the road features. We believe this to be a special case of the guided and/or multi-image (parallel) trackers.
- Coarse-Grained tracker — A tracker that takes "large" steps to extract the next set of road points. One example of such a tracker would be one that correlates some number of sequences of profiles in a single "step". Another would be a tracker that attempts to find parallel lines on each step.
- Verification tracker — A specific path is provided along which the tracker is required to walk. Some estimate of the goodness of the path is to be computed.

Each of these control scenarios presents unique implementation challenges . All currently fit cleanly within our composable tracker design model.

Thus far, we have used our composable tracker model to implement a simple cooperative road tracker. The implementation contains three trackers: a typical correlation-based tracker, a correlation-based tracker using probabilistic scoring, and a cooperative tracker that controls the correlation trackers. The subordinate trackers operate independently, with the cooperative tracker synchronizing their steps.

Given manually generated road starting points, results from this tracker are shown in Figure 6. Visually, one can see that the major road areas are almost completely tracked. There are fewer breaks in the longer road segments, and there are fewer overshoots at the ends of road segments. We continue to have problems in the suburban housing areas, and this is largely due to dense, overhanging vegetation.

The performance analysis numbers are summarized in Table 2. Comparing these numbers to those in Table 1, we can see that we have improved in almost every category. We have lowered the branching factor, quantitatively showing that there are fewer overshoots and mistracks. The quality and % correct numbers are slightly higher. Notably, the amount of ground truth explained (covered) has more than doubled. This is because the composable tracker does a better job of robustly tracking the given roads. The % redundancy is the one category where results have worsened. We expected this result because our current system includes overlap detection, and new composable tracking system does not yet have this capability. Our current efforts are focused on completing the implementation of the composable tracker framework, as well as on further improving it's performance.

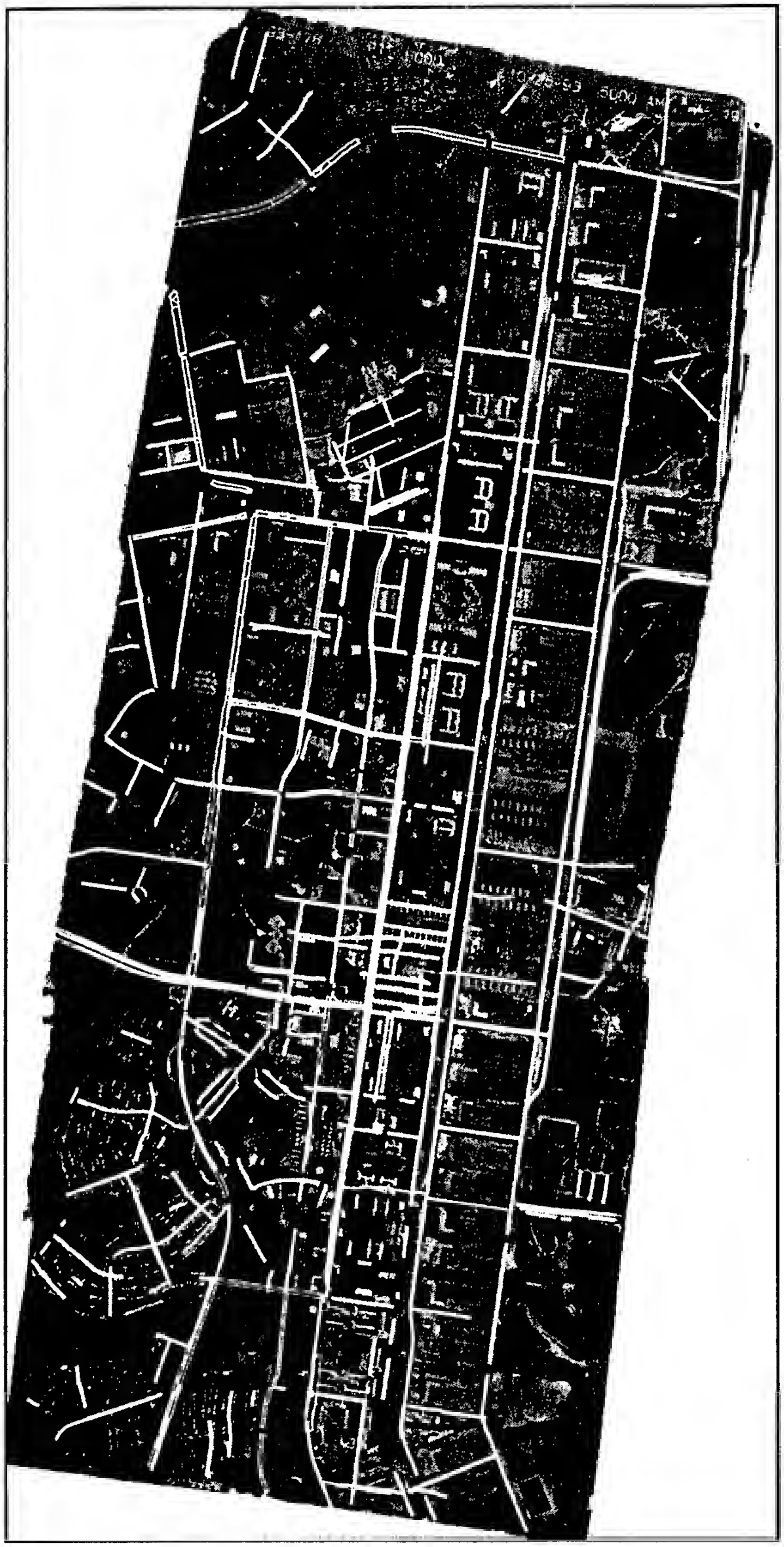


Figure 6: An orthographic view of Fort Hood, overlayed with the automated results generated by our object-space composable tracker. Manual starting points were used so that the performance of the tracker could be measured independent of detection errors.

Table 1: Performance evaluation measures generated using the current image-space automated road tracking system.

Image	Branching Factor	Quality Factor	% GT Explained	% Correct Hypotheses	% Redundant Pixels
FHN711	0.32	0.54	23.61	75.93	15.77
FHN713	0.55	0.48	37.91	64.55	27.61
FHN715	0.50	0.51	26.55	66.67	28.57
FHN717	0.35	0.60	34.87	74.34	27.55
FHN719	0.32	0.53	27.83	75.59	20.19

Table 2: Performance evaluation measures for the new object-space based composable tracker.

Image	Branching Factor	Quality Factor	% GT Explained	% Correct Hypotheses	% Redundant Pixels
FHN711	0.20	0.60	58.79	83.08	46.77
FHN713	0.25	0.58	50.86	79.82	44.95
FHN715	0.28	0.62	72.82	77.88	68.14
FHN717	0.24	0.68	70.01	80.45	57.73
FHN719	0.26	0.57	64.36	79.65	62.79

4 Experiments with Data Fusion

A common theme throughout MAPSLab research has been the belief that no individual computer vision technique can reliably provide a complete scene reconstruction. Thus, to achieve good performance, we need to gather a variety of information, extracted by various processes from multiple imagery of the area of interest. Then we need to synthesize this disparate information into a consistent model.

In three-dimensional scene analysis, the goal is to generate an interpretation of the scene that is as close as possible to the actual scene imaged. Such an interpretation can include the delineations and heights of buildings, the centerline and width of roads in a transportation network, a digital elevation model, and the segmentation and classification of the scene by surface material.

The key issue is the integration of many different sources of partial information. This problem appears under different guises: for example, given a set of different scene descriptions generated from a single image using a variety of techniques, how does one intelligently combine such partial information? The introduction of additional sensor types, temporal imagery, and multiple-look imagery create dimensions along which information fusion must be performed; as such, the complexity of the problem can increase. In some cases, increased amounts of data provide improved information. This may not necessarily follow, however. Complex systems having different sources of error may not reinforce correct partial interpretations nor refute incorrect ones. In addition, the results from different sets of imagery need to be represented in a common coordinate system in order to perform any fusion whatsoever.

4.1 Sources of information

The purpose of our fusion research is to determine how best to integrate disparate sources of information and how best to use the combined data to facilitate three-dimensional scene analysis. We are currently working four principle sources of information abstracted from near-nadir and oblique imagery over the area of interest. They are building hypotheses, road networks, elevation models generated from stereo matching and surface material classifications generated from multispectral and hyperspectral imagery.

As described in Section 2.1, the PIVOT system generates building hypotheses from the analysis of single or "monocular" images. The key advantages of this data is that the edges of objects are well localized and the vertical edges in oblique views yield good relative height estimates. However, with near-nadir imagery, poor height estimates may be generated due to lack of verticals. Also, edge breaks cause partial or total building loss (false negatives) and trees, cars and grassy quads can cause false positives.

As described in Section 3 a combination of the output of multiple road trackers which is combined to generate the road segments. The road segments are then linked to form intersections and full road networks. This road network segments a build-up area into logical areas for analysis, which can be used to control the application of other processes. However, the common occurrence of trees along the sides of the road can make tracking difficult in some areas. Here, information about the occurrence of high vegetation near a suspected roadway (*e.g.*, obtained from stereo and surface classification) could be used to provide guidance.

The third source of information is the stereo analysis of the areas where there exists stereo coverage. We have semi-automated the USATEC Digital Photogrammetric Compilation Package (Idl_DPCP) [Norvelle, 1992; Norvelle, 1981] to generate multiple stereo interpretations of built-up areas which can be projected into a common coordinate system and combined to form a mosaicked surface. The good points of the stereo process is that it can generate good height estimates from nadir imagery and, for continuous terrain, the matching algorithm tracks the surface very well. The difficulty with the stereo is that errors often occur at depth discontinuities, and that the edges of objects are smoothed by the area-based process. Finally, the elevation estimate degrades with increasing obliquity.

The fourth information source the surface material classification the area (See section 5). We use both automated and supervised classification of high-resolution hyperspectral data collected by the HYDICE sensor to segment and classify the surface according to material. This information is useful because common surfaces generally share the same surface materials, and the material classification of pixels adjacent to strong edges can be used to guide building hypotheses. However, the resolution of the data is generally less than that of the panchomatic images, and the airborne hyperspectral sensor results are difficult to geoposition.

There are alternative ways for organizing the three dimensional scene reconstruction threads into a combined processing approach. The basic division is either into a bottom-up (data directed) approach, where the results from the different methods are merged together; or, a top-down (knowledge-directed) approach, where the partial results are used to guide or select from other approaches. In the next two sections, we present our initial efforts in combining the partial results.

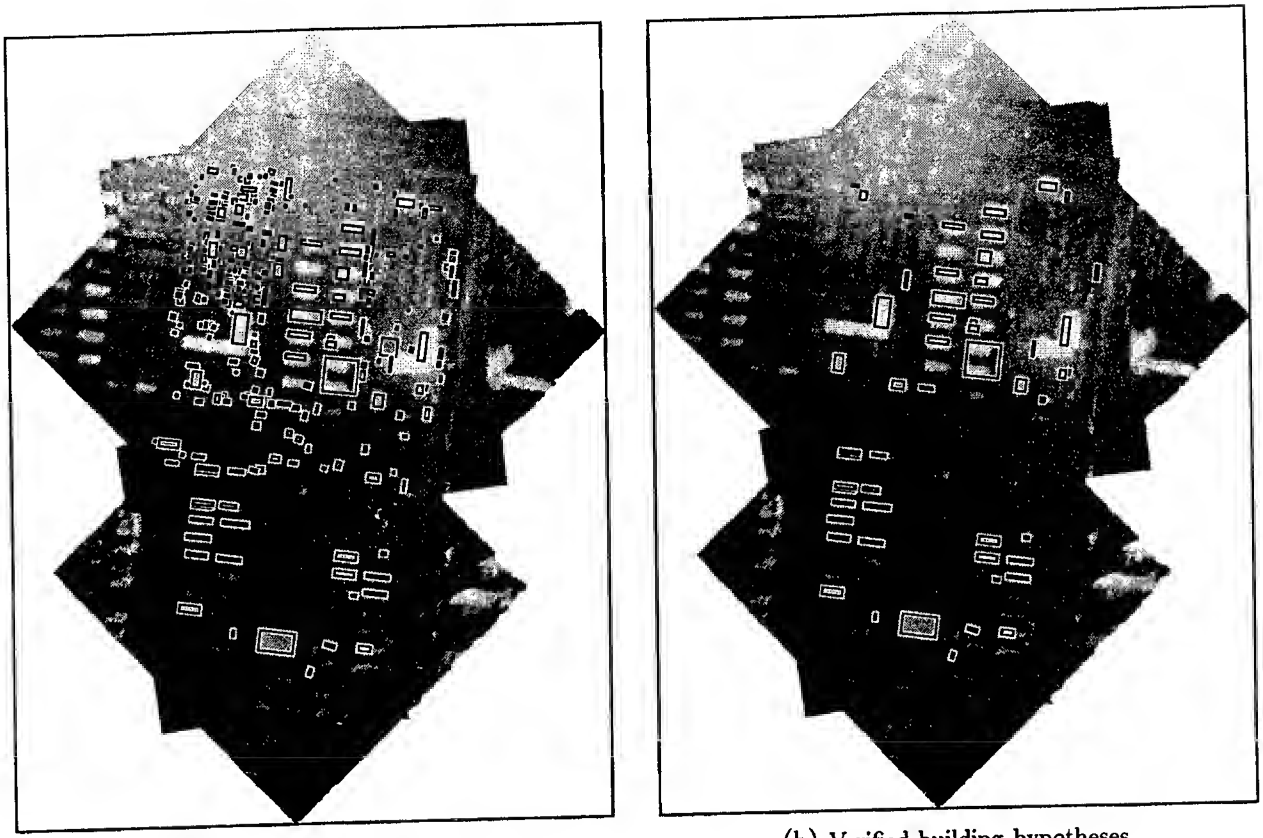
4.2 Bottom-up fusion

In this section we outline several ways that the results of the feature extraction/classification may be combined. It is difficult to imagine how to perform fusion without an *object space* framework within which to correlated information. Each of the following examples requires the choice of a common geospatial coordinate system, in our case we use the Universal Transverse Mercator system, since it is used by many-data sources and products. But the key notion is that image-based fusion, requiring projection of information into a particular image coordinate system is not flexible enough to support fusion from a variety of sensor or cartographic sources.

4.2.1 Stereo fusion

The stereo process may be applied to *multiple stereo pairs* of the same area. This process may be run both left-to-right and right-to-left on each pair of imagery to give a possible $2 \times \binom{n}{2}$ stereo results for n images. In practice, usually due to scale differences, some combinations are not good candidates for stereo matching. The resultant stereo results can be combined to reduce the blunders and reinforce actual elevation values. The stereo results shown in Figure 7 represent the averaged values from six stereo matches after outliers have been removed.

In combination with the stereo; building, road, individual tree, and tree canopy hypotheses may be used to remove incorrect stereo matches or to indicate areas where the stereo matching may be skipped due to obstruction. In addition, these areas may be marked for removal in support of the generation of a digital elevation model of the area. Likewise, buildings and roads may also be used as constraints in the stereo processing. The largest single problem in the stereo processing is the errors due to



(a) All building hypotheses.

(b) Verified building hypotheses.

Figure 7: Using stereo elevation to verify building hypotheses.

depth discontinuity and the associated occluded areas. Importing constraints to the stereo process at these locations should improve the resultant stereo matching.

4.2.2 Stereo and building hypotheses

The quality of stereo matching and monocular building extraction results appear to vary inversely to each other in their ability to estimate height from images of having different obliquity. By combining hypotheses from the two methods, the confidence of each height estimate may be weighted according to the obliquity of the original imagery.

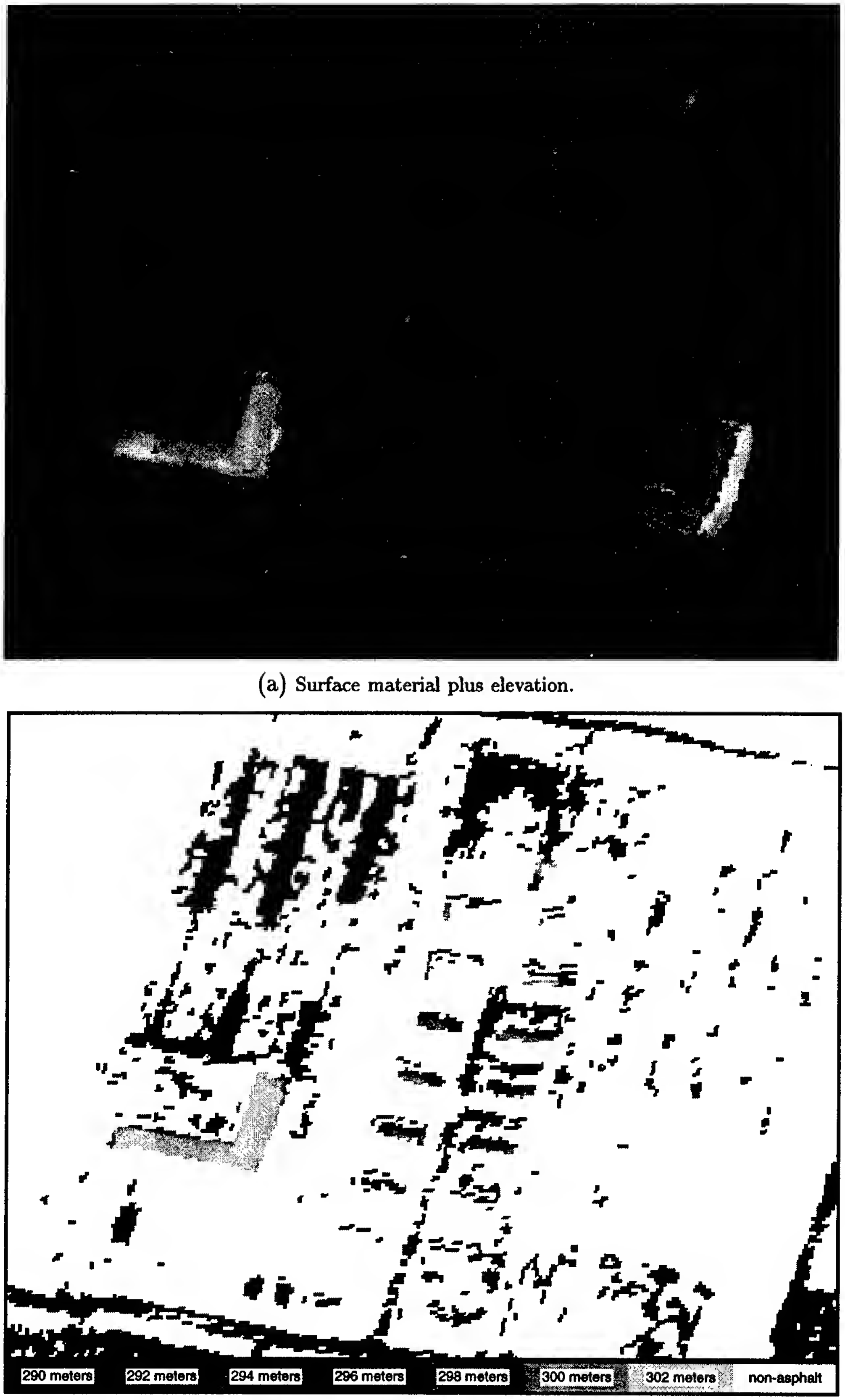
Stereo may be adjusted for the smoothing due to the correlation mask and used to verify or reject building hypotheses based on the difference of the inside and the surrounding height.

Figure 7a shows building hypotheses generated by PIVOT registered in object-space (UTM) with the elevation generated by the stereo processing.

Here those hypotheses that enclose raised areas and are surrounded by a surfaceat a lower elevation are considered likely candidates for buildings. In this case the criteria selected is that the interior is on the average of at least two meters higher than the surrounding ground after correction for the smoothing due to the stereo process. The verified hypotheses are shown in Figure 7b.

4.2.3 Stereo and multispectral surface material classification

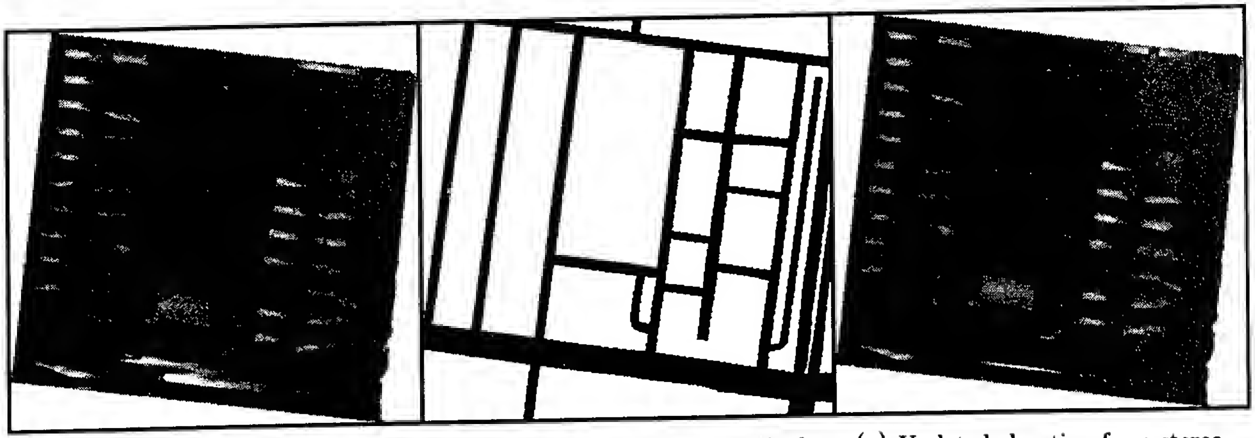
Figure 8 shows the initial overlay and fusion of the surface material classification with the height above the ground generated by the stereo processing. Here the value component of each material class is represented by the elevation. We can isolate raised areas representing trees and tree canopies as well as providing attribution to building features such as the asphalt and sheet metal roofs of the 'L'-shaped buildings to the left and right respectively. Figure 8b shows the asphalt component extracted from the combined elevation/material class image shown in Figure 8a. This shows clearly the raised asphalt (lighter) areas and the ground-level (darker) occurances.



(a) Surface material plus elevation.

(b) Asphalt surfaces, showing building roofs, parking lots and roadways.

Figure 8: Combining surface material classification with stereo elevation to identify materials by type and height.



(a) Original stereo elevation. (b) Road network from Idl_Woof. (c) Updated elevation from stereo.

Figure 9: Combining road tracks with stereo elevation to remove blunders due to moving vehicles.

4.2.4 Stereo and road networks

When roads are aligned with the epipolars of an image pair, then cars moving along the roads may be matched and their motion mistaken for elevation, with the cars moving one direction appearing at a higher elevation and those in the opposite direction appearing lower. The road network can be superimposed over the stereo and artifacts due to traffic along the road may be removed as shown in figure 9.

4.3 Top down fusion

The other fundamental approach to fusing data from different feature classification methods is to use one to focus or guide another. Here knowledge obtained about the scene may be projected to the image space or local coordinate system of the process to be supported.

For example, Figure 10 shows the areas which appear to be elevated relative to their local surroundings after analyzing the stereo results. Here, the stereo "blobs" have been projected back into the original image of the area where they may be used to either limit the search space by providing a focused areas-of-interest or to suggest early grouping of edges in the formation of building hypotheses.

This should help to improve performance both in terms of extraction quality (fewer false positives) and processing time (fewer features to consider).

5 Analysis of Hyperspectral Imagery: HYDICE

The low-to-moderate spatial resolution of multispectral data generally available has limited its usefulness to generating coarse descriptions of surface materials in fairly large, homogeneous areas. We believe that the high spatial resolution hyperspectral data experimentally available will allow us to generate surface material maps with dramatically better detail and fidelity, especially in urban areas. In addition, the higher spatial resolution of the surface material map will make it applicable to cartographic feature extraction—we will be able to see and classify parts of individual buildings or roads, thereby greatly improving our hypothesis verification capabilities. We have done preliminary work in using multispectral data with moderate spatial resolution in conjunction with high resolution panchromatic imagery; this has raised important issues in multisensor registration, cross-sensor information fusion, spatial-temporal differences, and in new techniques for automated material classification, as well as verifying the power of the fusion of such data.

The Naval Research Laboratory (NRL) Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor system is a 210 channel airborne scanner capable of providing the high spatial resolution imagery that we believe is required to support detailed spatial and spectral analysis. The sensor is mounted on a CV-580 aircraft; depending on aircraft altitude above ground level, the ground sample distance (GSD) varies from 0.75 to 3.75 meters. The HYDICE sensor is 320 pixels wide, giving a ground swath

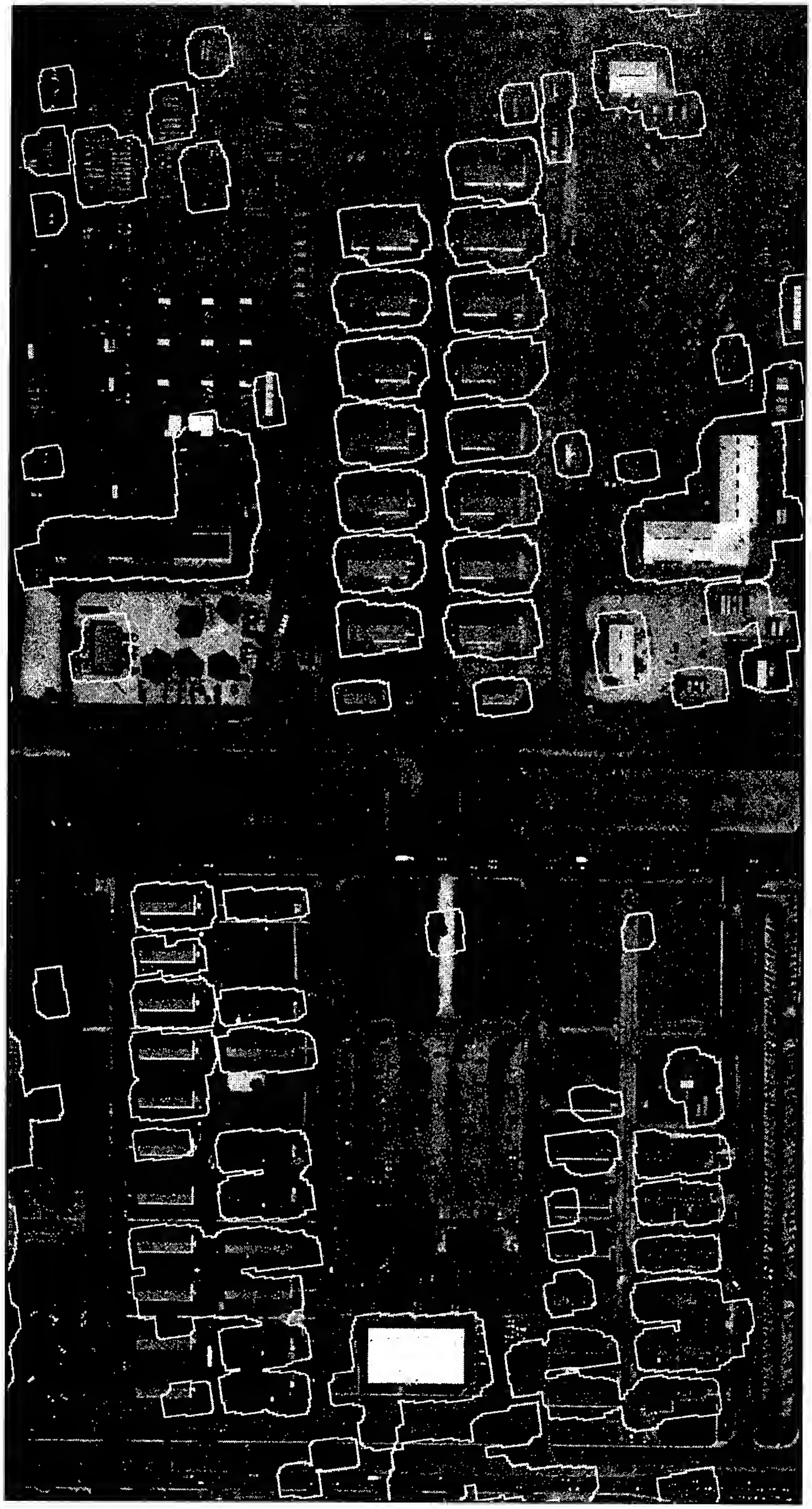


Figure 10: Stereo elevation may be used to localize searches.

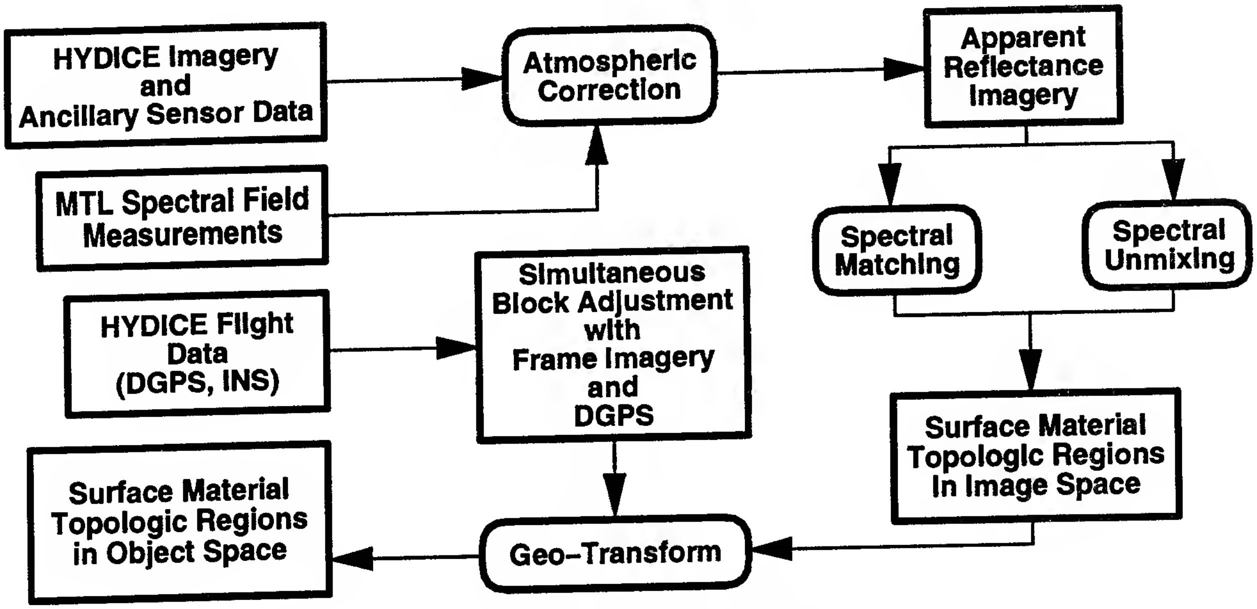

HYDICE Imagery and Ancillary Sensor Data
Atmospheric Correction
Apparent Reflectance Imagery
MTL Spectral Field Measurements
Spectral Matching
Spectral Unmixing
Simultaneous Block Adjustment with Frame Imagery and DGPS
HYDICE Flight Data (DGPS, INS)
Surface Material Topologic Regions In Image Space
Surface Material Topologic Regions in Object Space
Geo-Transform

Figure 11: Process model for HYDICE radiance imagery.

of 240 meters up to approximately a kilometer. Its spectral range extends from the visible to the short wave infrared (400 to 2500 nanometers) region, divided into 210 channels with nominal 10 nanometer bandwidths, varing from 7.6 to 14.9 nanometers, depending on channel location in the electromagnetic spectrum. The spectral bandpasses of three traditional multispectral imaging systems (Daedalus Airborne Thematic Mapper (ATM), Landsat Thematic Mapper (TM) and SPOT High Resolution Visible (HRV) Imaging Instrument) partially overlap the spectral resolution of the HYDICE sensor system.

The HYDICE sensor is, geometrically, a linear pushbroom sensor, which means that the CCD array is oriented perpendicular to the direction of flight and the 2D image is formed by platform motion (Figure 12). Physically, the sensor is an area array, with each array line perpendicular to the direction of flight recording the intensity in a different spectral band. Each image line consists of 320 pixels with an instantaneous field of view pixel of 0.5 milliradians, giving a total field of view of approximately 9 degrees. The first 7 and last 5 pixels in an image line are outside the optical path and contain no image data.

During October 24–27, 1995, we organized a hyperspectral data acquisition flown over Ft. Hood, Texas using the Naval Research Laboratory's (NRL) HYDICE sensor system to support research under our RCVW program. Additionally, natural color film was shot during the HYDICE collection flights and ground truth spectral measurements were acquired of surface materials to be imaged by the HYDICE sensor. A complete description of the data acquisition event as well as additional technical details are available [McKeown *et al.*, 1996b; McKeown *et al.*, 1996a].

Figure 11 gives an overview how we plan to process HYDICE imagery within the APGD research program. Two major components, radiometric and geo-position, must be addressed in order to effectively utilize the HYDICE imagery for information fusion in cartographic feature extraction. From the radiometric perspective, atmospheric conditions effect spectral scene illumination and spectral scene radiance reaching the HYDICE sensor. In order to compare surface material properties between flightlines, an atmospheric correction will be applied in order to convert HYDICE radiance imagery to apparent reflectance. This image conversion provides a framework to compare surface material properties across flightlines and to utilize spectral field measurements of surface materials for spectral analysis.

However in order to effectively make use of this information highly accurate geo-positioning is critical for fusing surface material topologic regions with information derived from our road and building feature extraction systems. Here the goals diverge somewhat from traditional remote sensing where precise registration is much less of an issue as aggregrate level descriptions of surface coverage or vegetation are acquired as much coarser spatial resolutions.

In Section 5.2 we describe research issues as well as pragmatic constraints in scene registration for the HYDICE sensor. In Section 5.2 we describe the process for radiometric calibration and preliminary material classification experiments performed using HYDICE imagery.

5.1 Registration of HYDICE imagery

Precise image registration is an absolute requirement for the fusion of information from multiple images. However, registration of HYDICE imagery presents special problems, due to its linear pushbroom imaging geometry and dynamic image formation. The HYDICE registration problem consists of two parts; modeling the perspective geometry of each single line, and modeling the platform motion along the flight line. Our approach to this problem uses navigation sensor information, frame imagery of the area, and geometric information within the scene to obtain the most rigorous and accurate registration possible.

The HYDICE sensor is mounted in a stabilized platform in a Convair 580 aircraft, along with other navigational and imaging sensors. The navigation equipment includes a GPS (Global Positioning System) receiver, capable of differential operation, and an inertial navigation system (INS). The readings from all these sensors are recorded on the data tape in real time.

Although a fairly complete suite of navigation sensors was available this does not, in general, solve the registration problem. Even in ideal cases, best accuracies of several to 10 meters on the ground are obtainable for the imaging geometries specified during our acquisition. When performing information fusion with imagery having a GSD of 0.3 meters, this accuracy is not adequate by itself and must be improved by a simultaneous solution. We were also limited by the lack of integration of the HYDICE and navigation sensors. HYDICE is an experimental system, with its main thrust being the development of high resolution spectral capabilities, so system integration to improve the positioning capabilities of the sensor has not been a priority. This greatly complicates the solution for the HYDICE imagery, since the weak imaging geometry of linear pushbroom sensors makes any navigation information very useful.

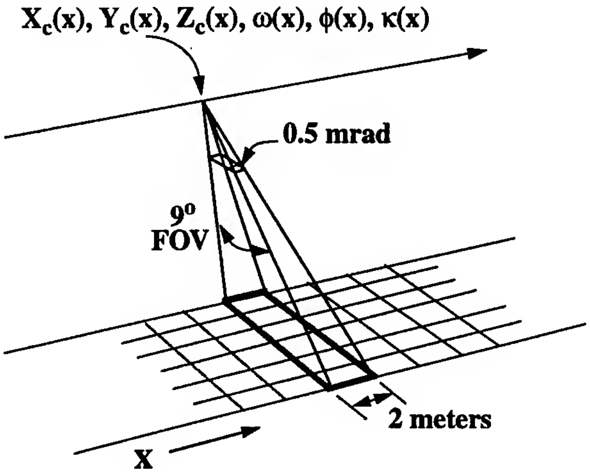


Figure 12: Linear pushbroom geometry.

We were also adversely impacted by specific acquisition conditions; due to time constraints, the Ft. Hood flights were made during turbulent atmospheric conditions which had adverse effects on the image geometry.

A linear pushbroom sensor can be thought of as a frame sensor with only one line in the x, or flight line, direction (Figure 12). The collinearity equations [McGlone, 1996], modified for use with linear pushbroom imagery, are:

$$\begin{aligned} \begin{bmatrix} U \\ V \\ W \end{bmatrix} &= M \begin{bmatrix} X_p - X_c \\ Y_p - Y_c \\ Z_p - Z_c \end{bmatrix} \\ 0 &= \frac{U}{W} \\ y - y_0 &= -f\frac{V}{W} \end{aligned} \tag{1}$$

5.1.1 The sensor and platform model

where the x coordinate is 0, y is the image coordinate and y_0 is the principal point along the sensor, f is the focal length, and X_p, Y_p, Z_p are cartesian world coordinates of the point, The position parameters, X_c, Y_c, Z_c, and the orientation parameters ω, ϕ, κ, (which determine the orientation matrix $M_{3,3}$) are functions of time, or equivalently, line number. To model this the value of each parameter $(X_c, Y_c, Z_c, \omega, \phi, \kappa)$ at a particular line is written as a polynomial function of line number x. The block adjustment solution then actually determines the polynomial coefficients, instead of the parameters themselves.

Not all of the six orientation parameters can be recovered in a resection solution, due to the sensor geometry. The linear sensor geometry means that the ϕ (pitch) angle will be highly correlated with position along the flight line, while the narrow field of view and lack of terrain relief means that the ω (roll) angle will be correlated with the cross-strip position. Without external information, such as angles or positions from navigation sensors, the ω and ϕ parameters must be held to 0 in the adjustment.

To prevent having to solve for high-order polynomials the flight line is divided into sections, with each section having its own set of polynomials. Continuity constraints are written at the section boundaries to insure that the orientation elements are identical at the boundaries and that calculated ground positions are consistent across the boundary.

5.1.2 Block Adjustment of HYDICE

Standard block adjustments use control and tie points to establish image positions and establish the relationships between images. While our block adjustment also uses discrete points, we have extended the solution to include object-space geometric constraints.

We make extensive use of straight line constraints, since we have a large number of object-space straight lines available over the motor pool area and these provide a large amount of geometric strength in determining the orientation parameters. Measuring a straight line extending across a number of image lines effectively provides a tie point in every image line in which the line appears. We also utilize right angle constraints which aid in determining the κ (yaw) angle of the sensor.

Three sets of imagery are used in this adjustment:

- The HYDICE imagery, collected in nine sidelapping flight lines. The imagery was flown at an altitude of approximately 4400 meters, giving a ground sample distance (GSD) of 2 meters.
- Color frame imagery, collected on the HYDICE flights with a KS–87 frame reconnaissance camera with a six inch focal length and a five inch format. Since this is an reconnaissance camera instead of a mapping camera, it has not been calibrated. The imagery was scanned at a 1 meter GSD.
- The RADIUS Ft. Hood imagery. This consists of about 40 nadir and oblique images, taken with a frame mapping camera and scanned at a GSD of 0.3 meters for the vertical images. These images have been previously block adjusted using surveyed ground control, and provide the basic geometric strength for the adjustment.

The control points for the adjustment were originally surveyed for the adjustment of the RADIUS images. Tie points are measured between the HYDICE, KS–87, and RADIUS images, along with straight lines and right angles.

5.1.3 Block adjustment procedure

The block adjustment is being performed using the object-oriented photogrammetry package described in [McGlone, 1992; McGlone, 1995], which allows the utilization of images with different geometries, along with the rigorous incorporation of geometric constraints.

We are performing the block adjustment sequentially, beginning with individual images, then sub-blocks, then the final block adjustment. Each individual HYDICE image is first adjusted using only the GPS information and holding the orientation parameters fixed. The next step is to adjust each HYDICE image along with its overlapping RADIUS and KS–87 images, using the measured control and tie points and the geometric (straight line and right angle) information. Finally we will perform sub-block adjustments with adjacent HYDICE images from the same flight line and from adjacent flight lines.

This procedure allows easier editing of input data for mistakes, since adequate redundancy is furnished by the frame imagery, while minimizing the solution time required since the larger adjustments are started with good orientation parameter estimates. This is an important consideration, since the final block will contain 48 HYDICE images, approximately 30 frame images, about 3000 constraints, and about 500 points. This amounts to approximately 10000 parameters.

5.1.4 Preliminary results

We have completed the GPS adjustment for all of the HYDICE images and have completed the solutions with point and geometric information, along with overlapping frame imagery, for two HYDICE images. We have used these solutions to understand the characteristics of the image data and the navigation data, and also to develop and optimize the data acquisition and solution procedures.

At this point, we have achieved registration accuracies for the HYDICE images of 5–10 meters (2–5 pixels). An orthoimage of image fhhydice4_3 is shown in Figure 13. This was produced from a solution dividing the image into four sections, using cubic polynomials for the X_c and Y_c positions and linear functions for the Z_c and κ parameters. The overall deformation of the imagery is shown by the curved edges of the orthoimage, while the removal of this general trend is evidenced by the fact that the roads are straight. At this level of accuracy, we are up against imaging distortions in the sensor and also the high-frequency oscillations ("wiggles") present in the imagery, which have an amplitude of about 2–3 pixels.

We have determined that the high-frequency oscillations cannot be modeled using orientation polynomials of reasonable degree, even when the strip is divided into very small sections. Introducing splines or equivalent models with sufficient degrees of freedom would add a very large number of unknowns to the solution. We are therefore investigating procedures to remove the high frequency deformations before solving for the overall strip geometry. We are still attempting to utilize the angular orientation information supplied, which would be the most direct method to remove the wiggles. In addition, we investigating image-based methods, using either line-to-line correlation or edge continuity, to determine the high-frequency deformations.

5.2 Analysis of HYDICE radiance imagery

Under our RCVW and APDG projects we have performed several initial experiments with the HYDICE Calibrated Radiance Imagery have which involved verification of radiometric calibration and classification of HYDICE-generated simulated Daedalus Airborne Thematic Mapper

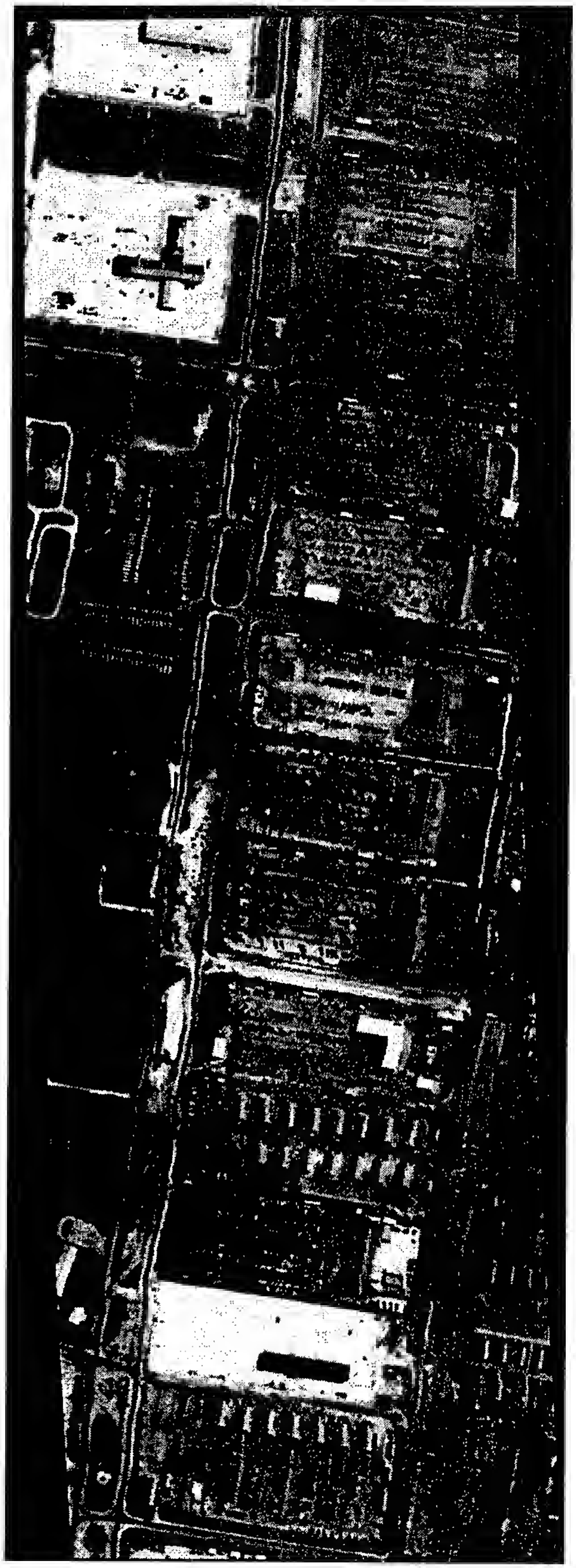


Figure 13: Orthoimage of FHHYDICE4_3.

(ATM) imagery. The following sections describe these activities.

5.2.1 Radiometric calibration

To verify the radiometric calibration of the HYDICE imagery, we examined the correlation between HYDICE imagery and MTL ground truth data by plotting spectral radiance curves of calibration panels in the calibration panel imagery of HYDICE Flightlines 4 and 5. For each gray level panel, we computed an average spectral radiance curve from the HYDICE imagery and compared it to a spectral radiance curve calculated from the MTL ground truth data. MTL provided ground truth spectral reflectance measurements for the calibration panels comprised of 8 to 12 measurements per gray level panel. The average spectral reflectance curve for each gray level panel was computed and multiplied by the spectral downwelling radiance collected by MTL during each HYDICE overflight.

This procedure was applied to Flightlines 4 and 5 panel imagery producing comparisons of the average spectral radiance per gray level panel between the HYDICE measurements and the computed MTL ground truth data. The spectral radiance comparisons for Flightlines 4 and 5 appeared quite good. The most significant differences are in the visible region and can be attributed to the atmospheric effects and upwelling radiance not present in the MTL ground truth data. This comparison work also lays the groundwork for conversion of HYDICE radiance imagery to apparent reflectance, allowing direct comparison and matching of HYDICE spectral apparent reflectance curves with MTL measured spectral reflectance curves.

5.2.2 Classification experiments

Due to the volume of image data generated by the HYDICE hyperspectral sensor, data reduction methods are being pursued. A first attempt at data reduction involves the averaging of HYDICE bands to simulate Daedalus Airborne Thematic Mapper (ATM) imagery. Figure 14a shows a near infrared band image from a portion of HYDICE Flightline 4 (FHDAED4_3) using band 7 from the simulated Daedalus ATM imagery.

Table 3: Surface material classes.

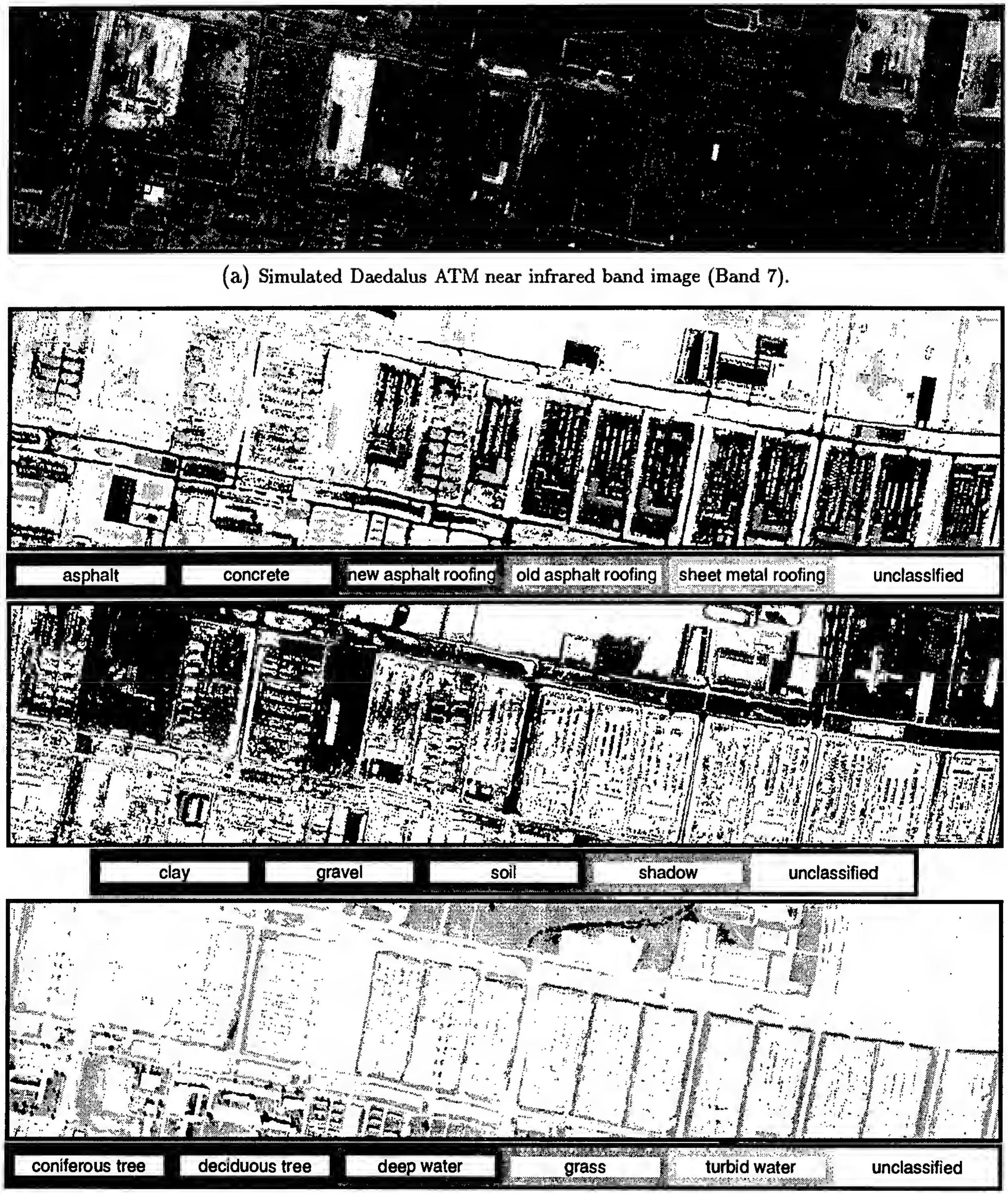
asphalt	gravel
clay	new asphalt roofing
concrete	old asphalt roofing
coniferous tree	shadow
deciduous tree	sheet metal roofing
deep water	soil
grass	turbid water

Using existing multispectral software, surface material classmaps are generated with a maximum likelihood classifier for the surface materials listed in Table 3. The selected training sets to compute the above spectral class statistics originate from an earlier section in Flightline 4 (*i.e.*, FHDAED4_2). This flightline segment contains natural vegetation, bare soil and waterbody features along with barrack/motor pool areas very similar in scene content as (FHDAED4_3). A problem that arose involved singular covariance matrices for the waterbodies. Upon closer inspection of the waterbody training set statistics, the SWIR bands (*i.e.*, Daedalus ATM band 9 and 10) had zero mean and standard deviation entries. This condition for waterbodies can be explained upon closer inspection of the HYDICE imagery.

The HYDICE calibrated radiance imagery from which the simulated Daedalus imagery was generated contains zero radiance values over the waterbodies. These zero radiance values are a result of clipping negative radiance values encountered during the conversion of a HYDICE Band-Interleaved (BIL) image cube to CMU image format. Negative radiance values are a byproduct of the HYDICE post-flight data processing when performing the radiometric calibration of the HYDICE imagery [Aldrich *et al.*, 1996].

Upon visual inspection of the resulting classification shown in Figure 14b, vegetated (*e.g.*, grass and deciduous tree) and asphalt features were labelled quite well. With the inclusion of roofing materials, building rooftops of barracks and motor pool facilities are visible from the surrounding parking lot background. Confusion between soil, gravel, and concrete features is still an issue.

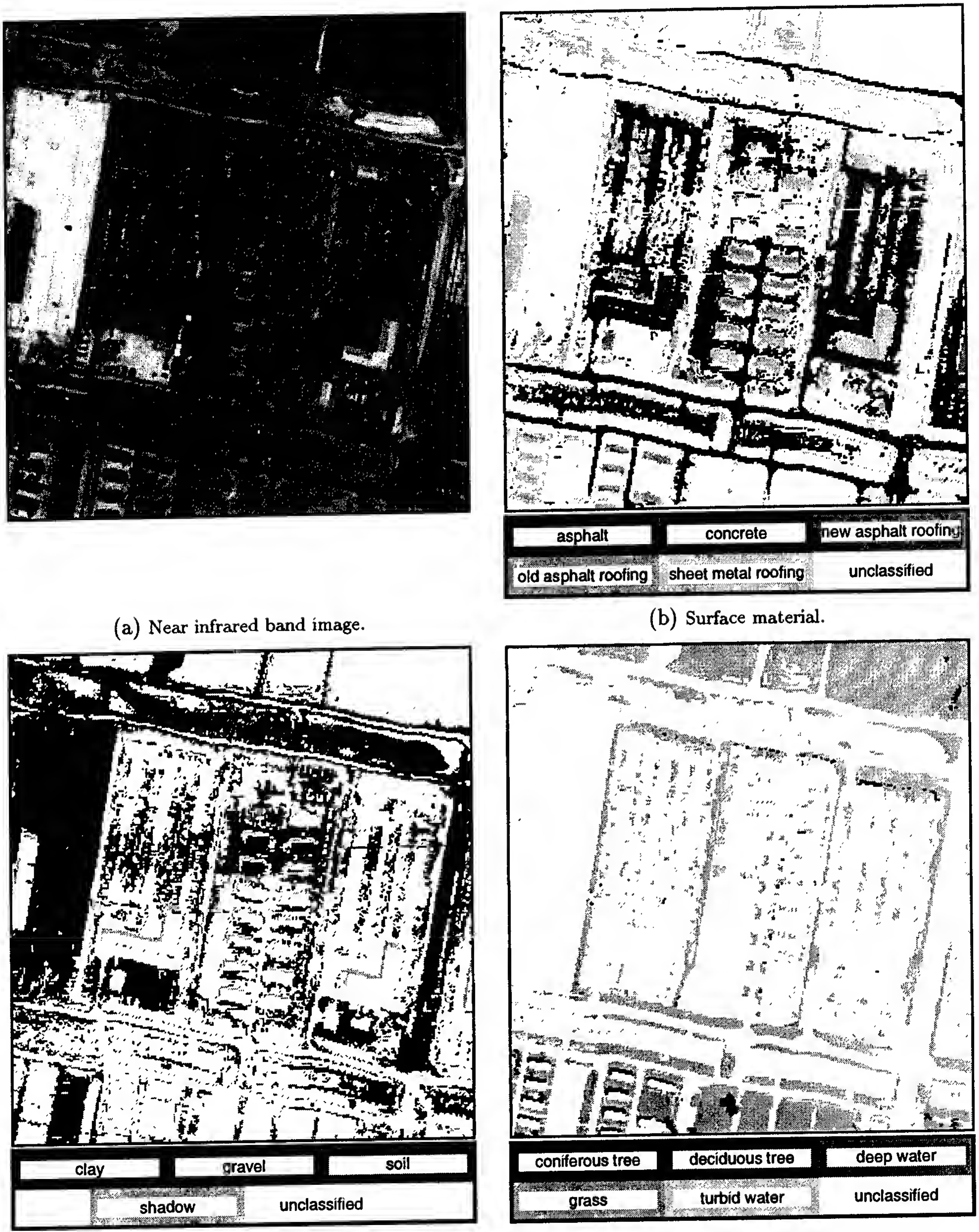
Figure 15a shows a close-up of test area RADT9 which is located left of center in Figure 14a. The resulting surface material classmap is shown in Figures 15b, 15c and 15d. The vegetated and



(a) Simulated Daedalus ATM near infrared band image (Band 7).

(b) Surface material classification.

Figure 14: Surface material classification from simulated Daedalus ATM image FHDAED4_3.



(a) Near infrared band image.

(b) Surface material.

(c) Surface material.

(d) Surface material.

Figure 15: Test area RADT9 surface material classification from simulated Daedalus ATM image FHDAED4_3.

parking lot areas along with asphalt road features are correctly classified while confusion between soil, gravel, and concrete features is still evident. An interesting effect of roof structure can be observed in the group of peaked roof buildings located in the center portion of Figure 14a. The peak roof structure affects the illumination intensity and type (*i.e.*, full sun versus full shade) incident on the roofing material. The change in spectral illumination effects the spectral content of these building roof structures as shown by their assignment to sheet metal roofing, old asphalt roofing or gravel classes.

6 Progress in Synthetic Environments

Constructing large-scale virtual world databases for ground-based simulation requires the integration of information from various sources, including digital map data, aerial and satellite imagery, detailed line drawings, and ground-based photography. Such virtual world databases have significant applications in DoD training, mission planning and rehearsal, and autonomous agent simulation.

Under the RCVW program we have focused on three aspects of database construction process. In Section 6.1 we discuss research in the preprocessing of existing spatial data to provide a coherent set of features to be modelled within the visual simulation. In Section 6.2 we describe progress in simulation database construction including Camp Pendleton and a revisit to the STOW-E terrain skin.

6.1 Efficient Simulation Database Representation

When planning the content of a simulation database, there are many constraints that can restrict the amount of spatial data that can be represented in the final compiled visual database. One of the first steps in database construction is the collection and preparation of the source geospatial data. The goal is to intelligently reduce the the geospatial source data while still retaining sufficient complexity to create a reasonably accurate and dense visual simulation database. The two main methods are selection, which will remove certain less important features, and generalization, which will reduce the number of points used to describe features while retaining the same basic shape. In this discussion we will focus mainly on road networks, but similar issues exist for vegetation coverage, drainage networks, coastlines, and built-up areas.

6.1.1 Selection

Selection of road networks can be accomplished in several ways. Currently we perform simple selection based on feature attribution. This can involve any arbitrarily complex criteria for examining the attributes, however, it is limited by those attributes available in the geospatial source data. In many cases, only attributes based upon intrinsic properties such as road width, and basic road type may be guaranteed to be available. Figures 16 and 17 show a simple example of selection of the major roads in the Camp Pendleton area using the *road width* attribute. We are working on more advanced solutions that interact with the database compilation process to use polygon density and network connectivity as selection criteria. Roads are weighted based on their importance, derived either from attributes or in terms of connectivity, and the less important roads are removed until the polygon density falls below a given set of budgets assigned for the visual simulation database.

6.1.2 Generalization

Once a road network has been selected, feature generalization is used to reduce the number of points needed to describe a road while retaining the same basic shape. This is particularly important since a road may be made up of an arbitrary number of intermediate points, few dictated by topology; some are artifacts of the feature compilation process. Our current implementation is based on the Douglas-Peucker distance based generalization algorithm. Each edge in the road can be given a different tolerance, or a default tolerance can be specified for the entire database. Common generalization algorithms perform on a feature by feature basis and this can create significant visualization problems when the associated polygonal representation is generated for the simulation database. For example, since portions of the feature edges can be moved, they can potentially intersect with other edges from other features. In dense high resolution road networks problems with proximity and point of closest approach are common.

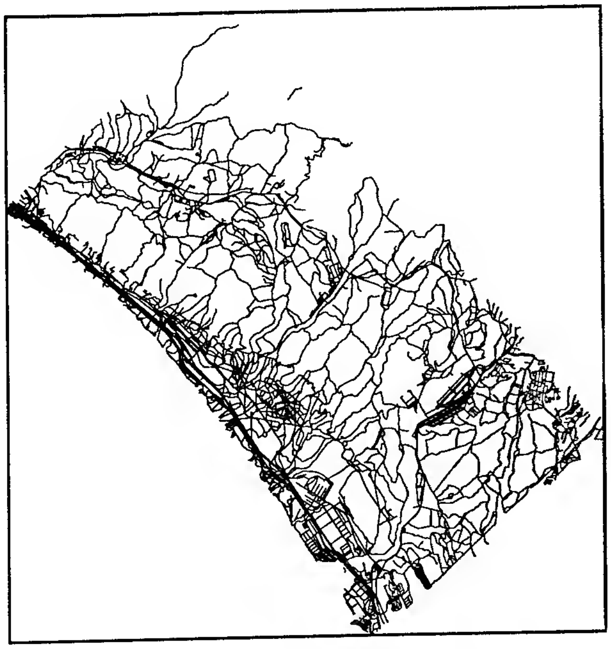


Figure 16: Original DTOP for Camp Pendleton.

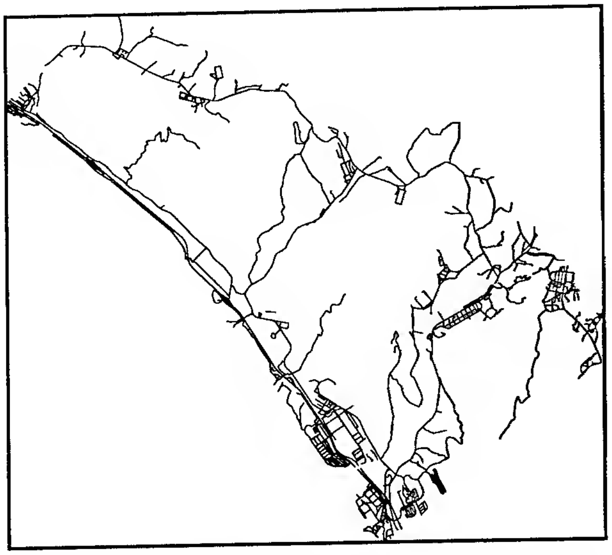


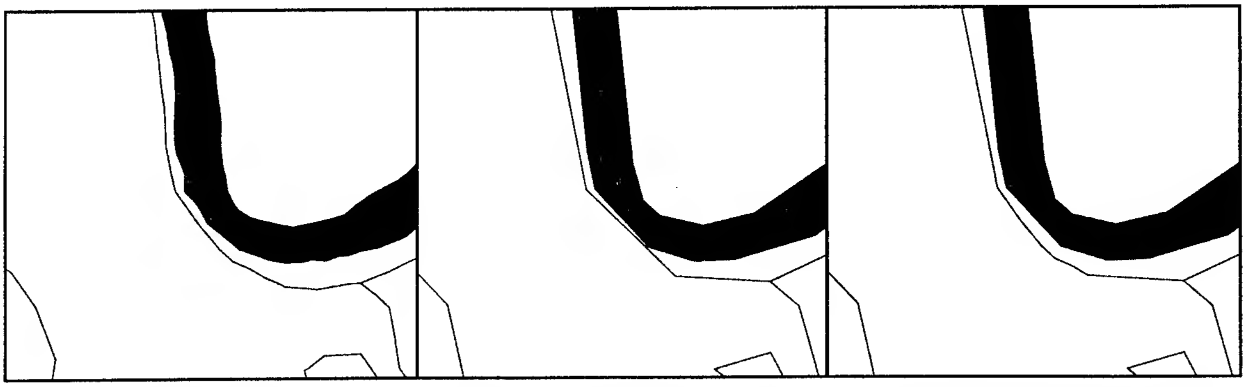
Figure 17: Selected roads (width at least 5 meters).

We have extended the basic generalization algorithm to prevent intersections from occuring. Additionally, a proximity distance can be specified, either on a per edge basis, or as a default value for all edges associated with a road feature. This prevents features from being generalized to be too close to another feature. Common examples occur when generalizing across traditional cartographic layers, such as transportation and drainage. Figure 18 illustrates the case where there is a road running along a river and generalization without proximity checking would create an unacceptable overlap in the polygonal representation of the road and riverbank. Another example where proximity checking is important is the case of a divided highway. The divided highway is especially interesting since in the cartographic representation, each collection of lanes is represented without any reference to the other. So, rather than handling this as a special case using feature type or attribution, it must be handled more generally at the geometric level.

The "best" generalization distance tolerance to use depends partly on the constraints of the visual simulation database (polygon limits, density, etc.), and partly on the cartographic compilation process that produced the feature data. Some source feature data has nearly straight edges with several extra points that can be removed without noticeably changing the edge, while other data must be changed significantly in order to remove the same percentage of points. Figure 19 shows three NIMA datasets containing road networks compiled at about the same cartographic scale. The number of roads and their extent is not relevant. What is interesting is that the point at which 50be removed requires generalization tolerances ranging from 4 meters to 13 meters. These observations lead to interesting research questions in data directed generalization rather than fixed thresholds based upon map scale or product type.

6.1.3 Matching/extraction

In some cases, it is desirable to use a selected and generalized dataset (Figure 20) as a template for selecting data from another dataset (Figure 16). For example, an old database might be rebuilt with new data or with a better generalization algorithm. Or we may simply want to compare cartographic data of the same area from two different sources. The basic procedure is to first create a corridor of a given width around each edge in the template dataset. Then, any edge in the test dataset that has a certain percentage of its length inside a corridor is kept. This produces useful initial results, and can be improved by changing the width and percent tolerance parameters. It does however, have some side effects. If the corridor width is too high or the tolerance too low, there will be a large number of short spur edges in the output. Likewise, if the width is too low, there will be gaps in some of the output edges. In general, it



(a) Original data.

(b) Generalized without proximity check.

(c) Generalized with proximity check.

Figure 18: Proximity check for generalization.

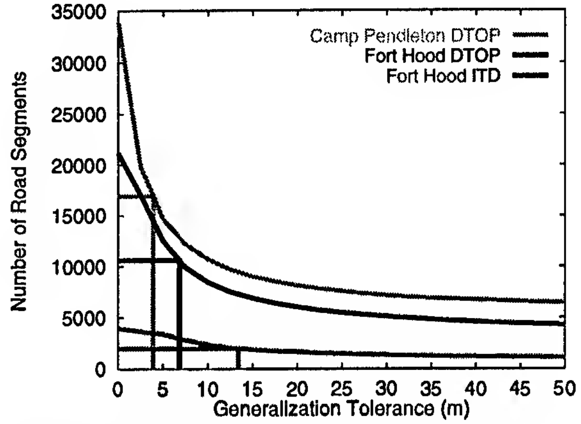


Figure 19: Tolerance needed to reduce data by half.

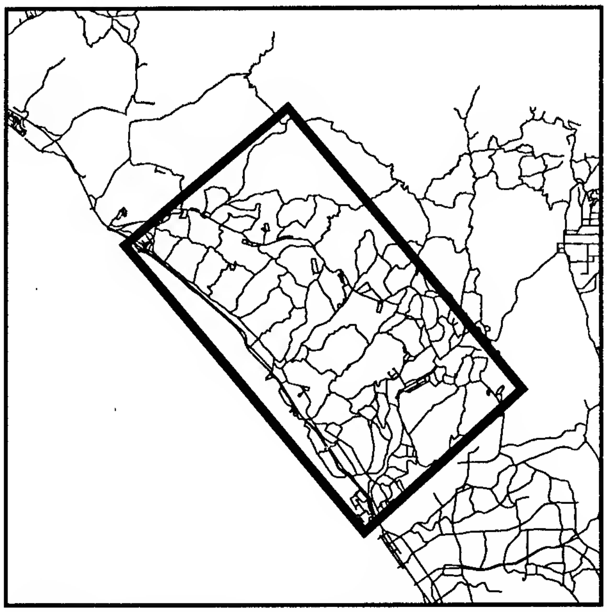


Figure 20: Selected, generalized from DTOP and TIGER.

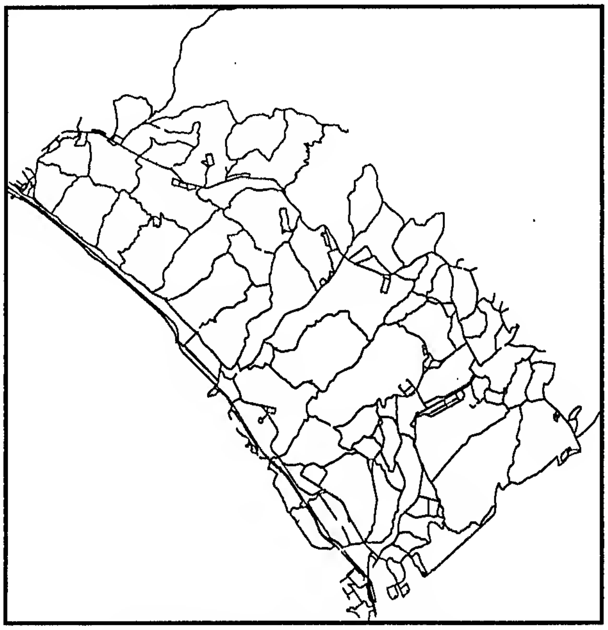


Figure 21: Extracted data.

seems to be preferable to have more spurs than gaps as they can be easily removed with some hand editing and simple automatic tests. Figure 21 shows the result of matching and extraction using the road network in Figure 20 as a template for extraction from the newer dataset in Figure 16.

Once extraction has been performed we can compare the resulting datasets for significant change. This is done by searching at points along an edge using one dataset as a reference. The result is a list of potential edge matches and the points of the edges can be further compared to find a common subset of points. Once the association is made the maximum distance between the edges and the total area between the edges can be computed. Of course, this comparison is limited by the spurs and gaps created by the extraction process which can significantly reduce the number of edges successfully matched.

6.2 Progress in Database Construction

Our early work in the area of automated simulation database construction involved the construction of Triangular Irregular Networks (TINs) that formed a simple but efficient bare earth terrain skin [Polis and McKeown, 1993]. This was closely followed by the development of integrated TINs (ITIN) that produces a highly efficent terrain skin which can incorporate roads, bridges, lakes, drainage features, including islands, directly into the terrain skin with minimal manual modeling. Initial efforts permitted roads to be automatically modeled to obey physical constraints with respect to road grade and side slope [Polis *et al.*, 1995]. ITINs have been shown to represent the character of the terrain significantly better than the right triangle representation used previously at the same polygon cost.

Recent developments have allowed for the experimental incorporation of tree canopies, arbitrary polygonal areas with reduced polygon budgets, and multi-level tins representing sea state and bathymetric surfaces. Drainage networks can be used in conjunction with a DEM to enforce streams that flow downhill and transportation networks can be used to generate cut-and-fill roads integrated into the terrain skin of various widths and road types.

While many experimental databases have been generated, it is notable that the MAPSLab ITIN process has been used in the development of several key STOW environmental databases, namely STOW-E (Central Germany), Range 400, and Camp Pendleton. In addition the Prairie Warrior exercise database (Chorwan), Fort Benning MOBA test database, and the Operation Kirby database were constructed using these tools.

6.2.1 Camp Pendleton database

During early 1996, a 40km × 40km integrated terrain skin of Camp Pendleton was generated to support and visualize Semi-Automated Forces (SAF) exercises. The Camp Pendleton exercise database was constructed using a variety of automated, semi-automated, and manual methods. We will briefly describe the database and our role in constructing it. A more complete description is available in [Polis *et al.*, 1997].

Terrain was modeled with ITINs that included bathymetry as well as terrain elevation data from six sources. Built-up areas and areas of high fidelity were accommodated as relaxed polygon budget constraints for the ITIN. Certain ground based features were integrated into the exercise terrain skin. They included drivable roads, simple bridges, lakes and rivers, underwater breaklines, an ocean surface, and detailed cliff structure along the Pacific coast.

The Camp Pendleton terrain database is supporting development of synthetic forces conducting amphibious operations within a Joint Task Force. Characterization of the littoral environment presents special challenges in developing advanced synthetic environments at the "seams" where air, land and sea intersect at the coastline and in the surf zone. The bulk of our work for this database was done over a three month period that ended with a delivery of the final integrated terrain skin to USATEC in the beginning of March, 1996. This included a visit to Camp Pendleton and a detailed tour of the significant beach structures in December, 1995.

A key requirement for Camp Pendleton was to model the cliff structure around several of the important beaches. However, structures like cliffs are not well reflected in DTED Level II due to a combination of sampling issues and interpolation in the construction. Since the cliffs represent an impediment to movement along the beach, we were forced to resort to hand modeling of the cliffs in the form of pairs of breaklines for the top and bottom of the cliffs. Figure 22 shows a photograph used as a reference during modeling and the corresponding view of the finished database. The resulting cliffs present an appropriate barrier to mobility.

6.2.2 Rebuilt STOW-E database

In 1994 we built the terrain skin for the database used in the STOW-E (Synthetic Theaters of War–Europe) demonstration. The database covered a 64 × 84 km area in Central Germany, and was created with an early version of the ITIN software. In that version, roads were the only feature type that could be

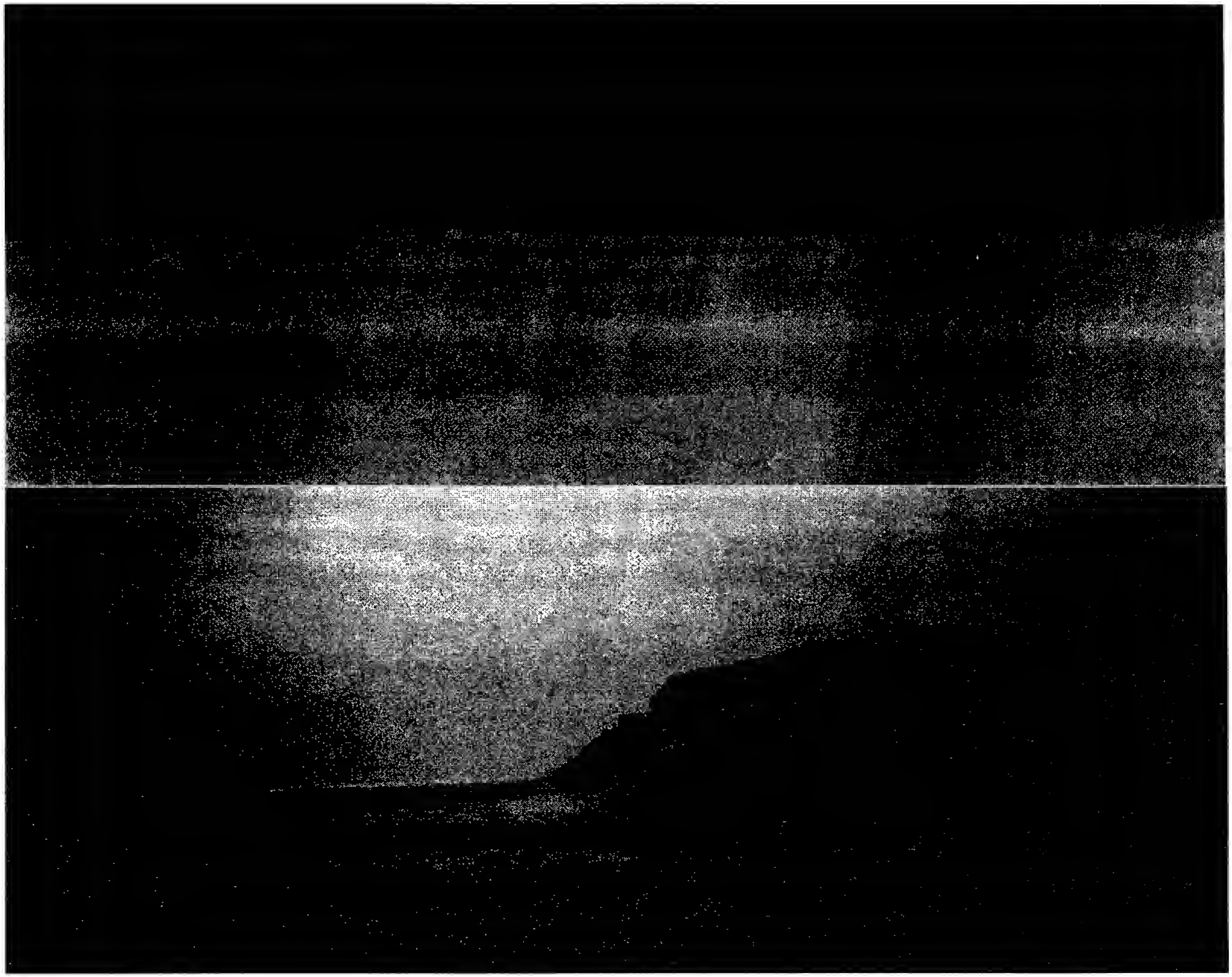
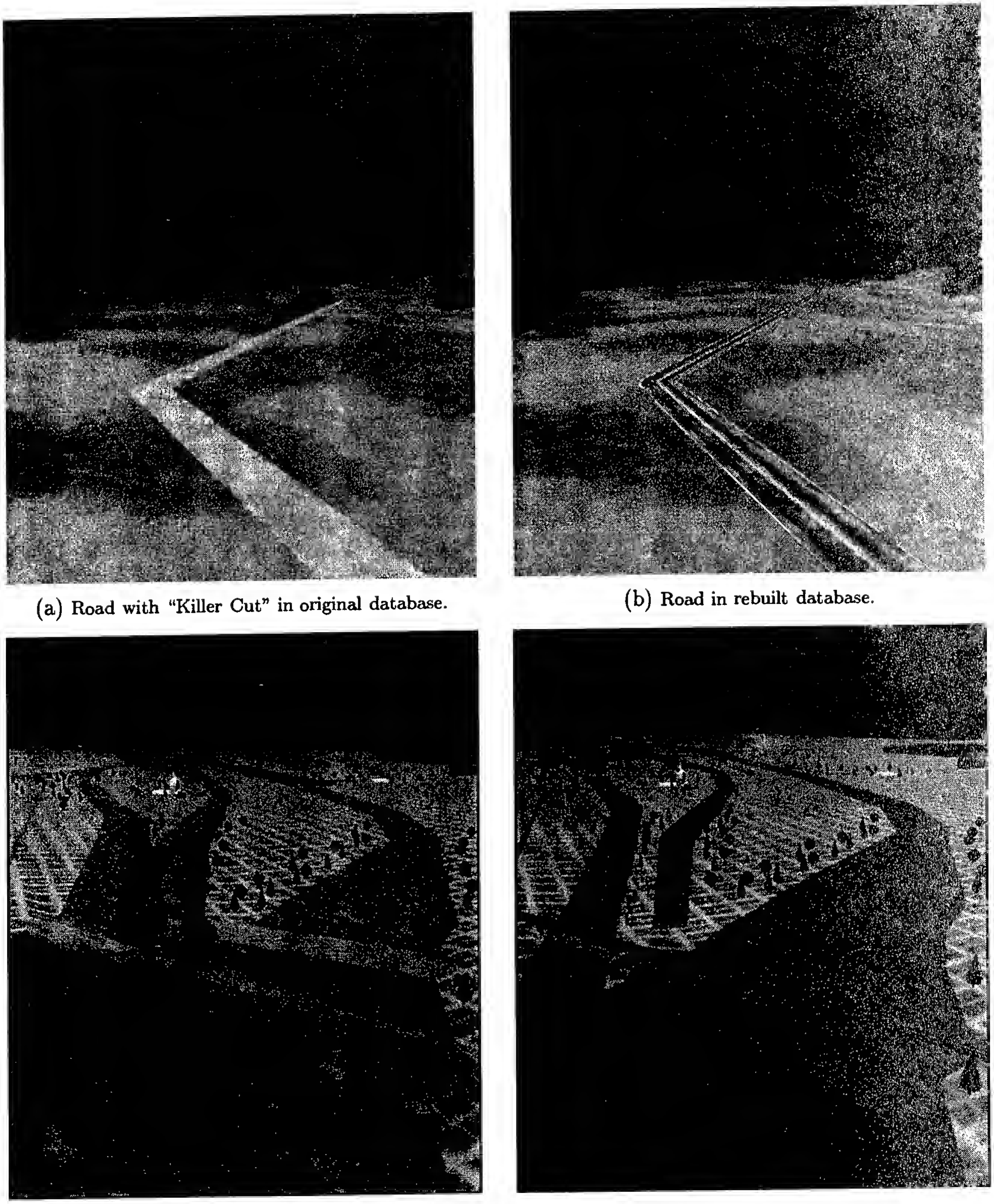


Figure 22: Cliff structure at White Beach, Camp Pendleton.



(a) Road with "Killer Cut" in original database.

(b) Road in rebuilt database.

(c) Regensburg in original database.

(d) Regensburg in rebuilt database.

Figure 23: Original and rebuilt STOW-E databases.

integrated. As a result, hand modeling was necessary for water bodies, bridges, and a variety of specialized features. In addition, feature integration wasn't always done smoothly, resulting in "killer cuts" (see Figure 23a). A vehicle traveling cross country and approaching the road from the left has only a few seconds warning of the steep drop. In some cases, the terrain height and texture are the same on both sides of the cut, making it even harder to recognize the sudden drop ahead. It is unrealistic for there to be no warning, and in fact the source data is not detailed enough to even know whether there really was a road cut. This, the cut was more of an artifact of the integration process than a carefully constructed model based on knowledge of the real world.

The ITIN system has been improved greatly since the original STOW-E database was built. As shown in the Camp Pendleton database, water bodies can be now integrated, and bridges are constructed automatically where roads cross water. Linear drainage features can be widened and integrated into the terrain using a process similar that used for roads. All of these features are integrated more smoothly into the terrain, virtually eliminating killer cuts. We can now construct a fully populated databases containing object models, not just a basic terrain skin. Tree canopies are generated and we can import models of individual trees, buildings, and other features and place them on the terrain. These improvements demonstrate the extent to which it is possible to rebuild the STOW-E database without resorting to hand modeling.

Figure 23 shows two views each of the original and rebuilt databases. Figure 23a shows a killer cut in the original STOW-E database. This was automatically corrected by running the road over the hill instead of through it. The result is shown in Figure 23b. Figures 23c and 23d compare views of Regensburg, a large city on the Danube. This area was extensively hand modeled, so it is a difficult challenge for the software. The bridges and complex river were successfully recreated and, in fact, the automatically generated bridge is neater than the hand modeled one. Models recovered from the original database were placed on top of the new terrain skin. In the upper left, one of the tree canopies generated by our software is visible.

Several discrepancies remain. The most obvious is the missing railroad bridge in the foreground. Since we do not yet integrate railroads, we did not construct railroad bridges. Another obvious difference is the area just left of image center which has a standard ground texture in the original database, but has a built-up area texture in the rebuilt database. This is because the built-up area texture was applied automatically to all polygons identified as built-up areas in the cartographic source data. It is not known what additional information was originally used to manually determine that the standard ground texture was appropriate. Perhaps the area was known to be a park from photographs or a visit to the site.

Acknowledgments

The research reported in this overview paper was supported by the Defense Advanced Research Projects Agency (DARPA/ISO/SE) and the U.S. Army Topographic Engineering Center under Contract DACA76-95-C-0009. Student research and training support was provided by the U.S. Army Research Office ASSERT program under Contract DAAH04-93-G-0092.

We particularly acknowledge the technical support of Mr. Doug Caldwell, USATEC, who has provided significant guidance, feedback, and direction to our overall program of research. Mr. George Lukes, DARPA/ISO/SE, Dr. Young Suk Sull, NIMA/TMPO, and Mr. Paul Foley, DMSO have contributed by focusing our database generation activities into realistic and externally useful directions. Dr. Thomas Strat DARPA/ISO/IU has been supportive of our research under the APGD contract.

We also thank the unsung hacker-youth of the Digital Mapping Laboratory for their contributions to various aspects of the research reported in this paper: Erastus 'Umbrella' Allen, Michael 'Frosh1' Fortson, Andrew 'Frosh2' Moise, Patrick 'Coach' Riley, Erik 'PhotoShop' Royse, Russell 'Russ' Schaaf, Steven 'Stealth' Sprang, Aaron 'BX' Wald.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Topographic Engineering Center, the U.S. Army Research Office, the Defense Advanced Research Projects Agency, or of the United States Government.

References

[Aldrich *et al.*, 1996] W. S. Aldrich *et al.* Hydice post-flight data processing. In A. E. Iverson, editor, *Proc. SPIE: Algorithms for Multispectral and Hyperspectral Imagery II*, vol. 2758, pp. 354–363, Apr. 1996.

[Biederman, 1985] I. Biederman. Human image understanding: Recent research and a theory. *CVGIP*, 32:29–73, 1985.

[Hsieh, 1996a] Y. Hsieh. Design and evaluation of a semi-automated site modeling system. In *ARPA IUW*, Palm Springs, California, Feb. 1996. ARPA, Morgan Kaufmann.

[Hsieh, 1996b] Y. Hsieh. Sitecity: A semi-automated site modelling system. In *CVPR*, pp. 499–506, San Francisco, CA, June 18–20 1996.

[McGlone and Shufelt, 1994] J. C. McGlone and J. A. Shufelt. Projective and object space geometry for monocular building extraction. In *CVPR*, pp. 54–61, Seattle, WA, June 19–23 1994.

[McGlone, 1992] J. C. McGlone. Design and implementation of an object-oriented photogrammetric toolkit. In *Int. Archives of Photogrammetry and Remote Sensing*, vol. XXIX, B2, pp. 334–338, 1992.

[McGlone, 1995] C. McGlone. Bundle adjustment with object space constraints for site modeling. In *Proc. SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, vol. 2486, pp. 25–36, Apr. 1995.

[McGlone, 1996] C. McGlone. Sensor modeling in image registration. In *Digital Photogrammetry: An Addendum to the Manual of Photogrammetry*, pp. 115–123. American Society for Photogrammetry and Remote Sensing, 1996.

[McKeown and Denlinger, 1988] D. M. McKeown and J. L. Denlinger. Cooperative methods for road tracking in aerial imagery. In *CVPR*, pp. 662–672, Ann Arbor, MI, June 1988.

[McKeown and McGlone, 1993] D. McKeown and J. C. McGlone. Integration of photogrammetric cues into cartographic feature extraction. In *Proc. SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, vol. 1944, pp. 2–15, Sept. 1993.

[McKeown *et al.*, 1996a] D. M. McKeown, Jr. *et al.* Research in automated analysis of remotely sensed imagery: 1994–1995. Technical Report CMU–CS–96–101, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, Jan. 1996.

[McKeown *et al.*, 1996b] D. M. McKeown, Jr. *et al.* Research in automated analysis of remotely sensed imagery: 1994–1995. In *ARPA IUW*, Palm Springs, California, Feb. 1996. ARPA, Morgan Kaufmann.

[McKeown *et al.*, 1996c] D. M. McKeown, Jr. *et al.* Automatic cartographic feature extraction using photogrammetric principles. In *Digital Photogrammetry: An Addendum to the Manual of Photogrammetry*, pp. 195–211. American Society for Photogrammetry and Remote Sensing, 1996.

[McKeown, 1996] D. M. McKeown, Jr. Top ten lessons learned in automated cartography. Technical Report CMU–CS–96–110, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, Jan. 1996.

[Norvelle, 1981] F. R. Norvelle. Interactive digital correlation techniques for automatic compilation of elevation data. Technical Report No. ETL-0272, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, VA 22060, Oct. 1981.

[Norvelle, 1992] F. R. Norvelle. Stereo correlation: Window shaping and DEM corrections. *PERS*, 58(1):111–115, Jan. 1992.

[Polis and McKeown, 1993] M. F. Polis and D. M. McKeown, Jr. Issues in iterative TIN generation to support large scale simulations. In AUTOCARTO 11: *International Symposium on Computer Assisted Cartography*, pp. 267–277, Minneapolis, MN, Oct. 30–Nov. 1 1993.

[Polis *et al.*, 1995] M. F. Polis, S. J. Gifford and D. M. McKeown, Jr. Automating the construction of large scale virtual worlds. *IEEE Computer*, 28(7):57–65, July 1995.

[Polis *et al.*, 1997] M. Polis, S. Gifford and D. McKeown. Toward automatic compilation of terrain skins from standard products. Technical Report CMU–CS–97–117, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, Mar. 1997.

[Shufelt and McKeown, 1993] J. A. Shufelt and D. M. McKeown. Fusion of monocular cues to detect man-made structures in aerial imagery. *CVGIP*, 57(3):307–330, May 1993.

[Shufelt, 1996a] J. Shufelt. Exploiting photogrammetric methods for building extraction in aerial images. In *International Archives of Photogrammetry and Remote Sensing*, vol. XXXI, B6, S, pp. 74–79, 1996.

[Shufelt, 1996b] J. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. In *ARPA IUW*, pp. 1113–1132, Palm Springs, California, Feb. 1996. ARPA, Morgan Kaufmann.

[Shufelt, 1996c] J. Shufelt. *Projective Geometry and Photometry for Object Detection and Delineation.* PhD thesis, Computer Science Department, Carnegie Mellon University, available as Technical Report CMU–CS–96–164, July 29 1996.

[Shufelt, 1997] J. A. Shufelt. Geometric constraints on hypothesis generation for monocular building extraction. In *Proc. SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III*, vol. 3072, to appear, Apr. 1997.

[Zlotnick and Carnine, 1993] A. Zlotnick and P. D. Carnine. Finding road seeds in aerial images. *CVGIP*, 57(2):243–260, Mar. 1993.

Progress in Computer Vision at the University of Massachusetts[1]

Allen R. Hanson, Edward M. Riseman, and Howard Schultz

Computer Vision Laboratory
Computer Science Department
University of Massachusetts
Amherst, MA 01003
URL: http://vis-www.cs.umass.edu/

Abstract[1]

This report summarizes progress in image understanding research at the University of Massachusetts over the past year. Many of the individual efforts discussed in this paper are further developed in other papers in this proceedings. The summary is organized into several areas:

1. 3D Site Modeling from Aerial Views
2. Terrest Terrain Reconstruction System
3. Terrain Classification /Force Monitoring
4. Content-based Image Indexing
5. Learning in Vision
6. Miscellaneous Related Research

The research program at UMass has as one of its goals the integration of a diverse set of research efforts into systems that are ultimately intended to achieve robust, real-time image interpretation in a variety of vision applications.

I. 3D Site Modeling from Aerial Views

1.1. The Ascender Site Modeling System

Under the DARPA/ORD RADIUS program, UMass developed the ASCENDER system (Automated Site Construction, Extension, Detection and Refinement) for automatically populating a site model with 3D building models extracted from multiple, overlapping images (both nadir and oblique) of the site (Collins et al. 1996). The UMass design philosophy emphasizes model-directed processing, rigorous 3D photogrammetric camera models, and fusion of information across multiple images for increased accuracy and reliability. The Ascender system has been transferred to Lockheed-Martin and the National Exploitation Laboratory (NEL) where it has been evaluated on classified data sets.

The Ascender system acquires, extends and refines 3D geometric site models from aerial imagery with known parameters. To acquire a new site model, an automated building detector is run on one image to hypothesize potential building rooftops. Supporting evidence is located in other images via epipolar line segment matching in constrained search regions. The precise 3D shape and location of each building is then determined by multi-image triangulation, and shape optimization under constraints of 3D orthogonality, parallelness, colinearity and coplanarity of lines and surfaces. As new images of the site become available, they are matched to the partial site model and model extension and refinement procedures are performed to add previously unseen buildings and to improve the geometric accuracy of the existing building models. In this way, the system gradually accumulates evidence over time to make the site model more complete and more accurate.

Based on initial experience in the evaluation at the NEL, major changes have been made to Ascender's control system. The original system used a single reference image to generate roof hypotheses in the form of polygons, and then used the remaining images to verify/reject buildings by constructing a 3D model. If a building hypothesis was not found in the reference image, the building would not be constructed *even though* it might be clearly visible in one or more of the other images. A new control strategy has been implemented under which all images are processed uniformly; polygons found in *any* image are used as the set of initial rooftop hypotheses from which the 3D reconstruction begins.

Tests have been performed on a subregion of the Fort Hood dataset. Polygons were detected in seven images and redundant polygons eliminated on the basis of overlap. Each of the remaining

[1] This work was supported in part by DARPA under AO # E658 and DARPA contract numbers DAAL02-91-K-0047 (via ARL), DACA76-92-C-0041 (via TEC), F30602-94-C-0042 (via Rome Laboratory), by NSF grant number CDA-8922572 and by Lockheed Martin under Subcontract/PO number RRM072030.

polygons was then used to construct a 3D building model Models that had a side or height of less than 5 meters were eliminated. Using this scheme 92% of the 76 rooftop polygons were detected, leaving six polygons missed in al! seven images. An additional 45 polygons represented false positives from either errors in the 2D grouping process that survived verification or the reconstruction of a cultural feature other than a building (parking areas, playing fields, etc.) that had errors in height due to limited support from the image set.

I.2 Ascender II: Context Sensitive Control of Reconstruction

Work on the Ascender I system demonstrated that the use of multiple strategies and 3D information fusion can significantly extend the range of complex building types that can be reconstructed (Jaynes et al. 1996). Under the DARPA APGD program, we are designing and building the successor to the Ascender system. The design approach is based on the observation that while many IU techniques function reasonably under constrained conditions, no single IU method works well under all conditions. Consequently, work on Ascender II is focusing on the use of multiple alternative reconstruction strategies from which the most appropriate strategies are selected by the system based on the current context. In particular, the new system will utilize a wider set of algorithms that fuse 2D and 3D information and make use of EO, SAR, IFSAR, and multispectral imagery during the reconstruction process. We believe that such a system will be capable of more robust reconstruction of three dimensional site models than has been demonstrated in the past and will significantly reduce the effort required by image analysts during the reconstruction process. This in turn will result in faster development of topical situational and visualization products of military significance.

Ascender II is organized into two subsystems The IU components of the system responsible for manipulating image data are being constructed in the ARPA RCDE system, while the control and inferencing components are represented as Bayesian belief networks constructed using the Hugin system (Jensen 1996). Communication between the two subsystems is currently being supported by UNIX socket facilities using packets structured for this application. Control policies (strategies) are associated with the object classes represented in the network. Execution of a control policy results in the accumulation of evidence for/against the corresponding network node. This evidence is propagated through the network based on the Bayesian probability tables constructed as part of the knowledge base. Currently, a maximum uncertainty policy is used to select the next node in the network to be expanded, although more sophisticated mechanisms are being explored as part of the research. A second issue being examined relates to the appropriate granularity of the control policies and therefore the granularity of the reconstruction systems themselves. Finally, a major effort is underway to develop new and expanded reconstruction strategies and the IU procedures required to support them.

I.3 Reconstruction Strategies

Our focus under the DARPA APGD program will be on more general 3D reconstruction strategies that utilize multiple types of features (points, lines, surfaces), and that can be applied to a wide range of parameterized building classes. The system-level approach involves multiple alternative detection and reconstruction strategies, invoked by clear contextual cues, that combine a wider set of algorithms and features for generating and fusing 2D and 3D information. The strategies being developed will be utilize both monocular optical data as well as digital elevation data obtained from IFSAR or multi-view stereo reconstruction from EO data.

Several new reconstruction strategies are being developed for this system. To take a single example, Ascender I rooftop hypotheses from a single optical image are projected into registered elevation data and used to trigger the application of a rooftop model matching strategy to the restricted subset of the data. The model matcher uses a knowledge base of approximately 12 parameterized rooftop models (including flat, peaked, and curved roof models) and matches by correlating a histogram of surface orientations derived from the data with the orientation pattern of the model surfaces on the Gaussian sphere.

Initial experiments have been obtained with the Ascona/ISPRS "Flat Scene" [ftp://ftp.ifp.uni-stuttgart.de/pub/wg3/]. This scene contains several peaked roofs with different slopes and cluttered with gabled windows, chimneys, etc. In addition, the elevation data contains noise unavoidably introduced through the stereo reconstruction process. In this experiment, the top two models resulting from the correlation process were selected and fit to the constrained elevation data. The model with the lowest residual fit error was chosen for the final reconstruction. Excluding the six rooftops missed by the hypothesis generation phase, the remaining eleven rooftops were correctly classified as peaked roof buildings. After surface fitting the models using the initial parameters found during indexing, the average residual fit error was 0.192 meters. See Jaynes et al. (1997) in these proceedings for more detail.

1.4 Reconstruction Strategies from IFSAR

As part of an ORD-sponsored feasibility study, several approaches to bottom-up reconstruction of spatially coherent structures from IFSAR data have been explored. The strategies are being extended under the APGD program for inclusion in Ascender II. In one approach, building footprints extracted from optical data are used as a focus of attention mechanism to select subsets of the IFSAR data. This is followed by the application of robust 3D surface fitting techniques to the elevation data. In a second approach, variations on traditional region growing methods are applied to the IFSAR data alone. Geometric constraints can be imposed during region growing to produce rectangular or rectilinear shapes. These ideas have been explored using the initial Sandia/Kirtland AFB dataset; this dataset has since been characterized as being particularly noisy with a large number of drop-outs and outliers. This effort is described in more detail in (Hoepfner et al. 1997; Jaynes et al. 1997).

1.5 Surface Microstructure Extraction from Multiple Aerial Images

Building surfaces with microstructures provides important information for many military and civilian applications. The extraction of small scale information from aerial imagery is difficult due to problems caused by perspective distortion, data deficiencies, and shadows and occlusions. A subsystem has been developed for improved extraction of site model details, often at a scale close to the limits of image resolution.

An Orthographic Facet Image Library (OFIL) system and a generic window and door extraction module has been constructed under the assumption that an initial site model is available, and sufficient camera and light source information is known. The OFIL system is designed to systematically collect the building facet intensities from multiple aerial images into an organized orthographic library, eliminate the effects of shadows and occlusions, and combine the intensities from different sources to form a complete and consistent intensity representation for each facet. A 'Best Piece Representation' algorithm is designed to combine intensities from multiple views, resulting in a unique surface intensity representation. The window extraction module focuses attention on wall facets, attempting to extract the 2-D window and door patterns attached to the walls. The algorithms are typically useful in urban sites. Experiments show successful applications of this approach to site model refinement and improved fly-through scene visualizations; details can be found in (Wang et al. 1997).

2. Terrest Terrain Reconstruction System

The UMass Terrain Reconstruction System (Schultz 1994) deals effectively with highly oblique viewing conditions using a texture correlation algorithm that incorporates (1) hierarchical unwarping, (2) weighted cross-correlation and (3) narrow search subpixel registration.

Recent extensions of Terrest (Schultz, Stolle et al. 1997) to site modeling applications involve the incorporation of boundary constraints into the correlation masks. The windows of a correlation mask are restricted to be completely on either side of a boundary, thereby causing the mask to be adaptive to the context in which it is applied. For example, with building roofs, correlation masks are automatically shaped to lie entirely inside or outside the area where a polygonal boundary has been detected (Quam 1984). The net result is significantly sharpened digital elevation maps (DEMs).

2.2 Automated Bundle Adjustment

An effort is underway to automatically select image match points as a precursor to the bundle adjustment process necessary to precisely register images and to compute precise relative camera orientation. The approach we are developing uses building corners as the feature points of choice. These features are weighted according to their distinguishabilty and correlation masks are generated from features of high confidence. We are examining the question of whether or not the correlation peaks resulting from the use of these features during matching can be analyzed in order to distinguish between true correspondences and false matches.

Since the degree of precision obtained in the camera pose parameters depends strongly on the accuracy of the match point locations, a robust estimation technique is used to remove outliers, i.e. false correspondences, which passed the local tests. The final step is a least-squares technique to obtain the final relative orientation. Typically 20 to 30 distinctive points are required for accurate results.

3. Algorithms to Support Force Monitoring

3.1 Using Three-Dimensional Features to Improve Terrain Classification

Image texture has long been regarded as the spatial distribution of gray-level variation, and texture analysis has generally been confined to the 2- D image domain. We have demonstrated the utility of "3-D world texture" as a function of 3-D structures (Wang et al. 1997)and proposed a set of 3-D textural features. The proposed 3-D features have a great potential in terrain classification. Experiments have been carried out to compare the 3-D features with a traditional 2-D feature set. The results show that the 3-D features significantly outperform the 2-D features in terms of classification accuracy and training data reliability.

The classifications have been used to generate ground cover maps and a skeletal road network.

3.2 Visibility Analysis for Force Monitoring

Visibility analysis algorithms have been developed for a variety of force monitoring scenarios, including stealth path planning, placing a set of observers on an elevation map to maximize spatial coverage, and for analyzing when/where a force of a given size would be detected over a given line of advance.

Our theoretical work is based on the Art Gallery Problem, which is the problem of determining the number of observers necessary to cover an art gallery such that every point is seen by at least one observer. A polynomial time solution has been developed for the 3-D version of the Art Gallery Problem. Because the problem is NP-hard, the solution presented is an approximation, and bounds to the solution are presented. Our solution uses techniques from computational geometry, Graph coloring and set coverage. A complexity analysis for each step and an analysis of the overall quality of the solution has been derived (Marengoni et al. 1996). This general algorithm has been applied to several problems in visibility analysis on an elevation map.

4. Content-based Image Indexing

4.1 Center for Intelligent Information Retrieval (CIIR)

The Center for Intelligent Information Retrieval (CIIR) conducts leading basic research in the area of information systems. This national center is one of only four centers in science and engineering to be funded in 1992 by the National Science Foundation under its State/Industry University Cooperative Research Centers program. One of the goals of the CIIR is to develop tools that provide effective and efficient access to large, heterogeneous, distributed, text and multimedia databases. A new partnership between the Computer Vision Lab and CIIR is focused on content-based multimedia indexing and retrieval, a difficult yet vitally important task. The aim of content-based retrieval is to efficiently find images which contain the object represented in a query image in a large database.

4.2 Appearance-Based Indexing & Retrieval

A system to retrieve images using a syntactic description of appearance has been developed and appears in these proceedings (Ravela and Manmatha 1997). A multi-scale invariant vector representation of images in the database is obtained by first filtering with Gaussian derivative filters at several scales and then computing low order differential invariants; this done off-line.

Run-time queries are designed by the users from an example image by selecting a set of salient regions. The responses corresponding to these regions are matched with those of the database and a measure of fitness per image in the database is computed in both feature space and coordinate space. The results are then displayed to the user sorted by the match score. From experiments conducted with over 1500 images it has been shown that images similar in appearance, and whose viewpoints are within 25 degrees of the query image, can be effectively retrieved.

4.3 Color-Based Indexing & Retrieval

A new multi-phase, color-based image retrieval system (FOCUS) has been developed which is capable of identifying multi-colored query objects in an image in the presence of significant, interfering backgrounds. The query object may occur in arbitrary sizes, orientations and locations in the database images. The color features used to describe an image have been developed based on the need for speed in matching and ease of computation on complex images while maintaining the scale and rotation invariance properties. The first phase matches the color content of an image with the query object colors using an efficient indexing mechanism. The second phase matches the spatial relationships between color regions in the image with the query using a spatial proximity graph (SPG) structure designed for the purpose. The method is fast and has low storage overhead. Test results with multi-colored query objects from man-made and natural domains show that FOCUS is quite effective in handling interfering backgrounds and large variations in scale (Das and Riseman 1997).

4.4 Text Detection & Extraction in Images

There are many applications in which the automatic detection and recognition of text embedded in images is useful. These applications include multimedia systems, digital libraries, and Geographical Information Systems. However, text is often printed against shaded or textured backgrounds or is embedded in images. Examples include maps, advertisements, photographs, videos, and stock certificates. Current OCR and other document recognition technology cannot handle these situations well.

A four-step system has been developed that automatically detects and extracts text from images by treating it as a texture (Wu et al. 1997). Potential text locations are found by filtering second-order derivatives of Gaussians at three different scales. Second, vertical strokes from horizontally aligned text regions are extracted. Based on several heuristics, such as height similarity, spacing and alignment, strokes are grouped into tight rectangular bounding boxes around text strings. These steps are then applied to a pyramid of images generated from the input images in order to detect text over a wide range of font sizes, and then the boxes are fused at the original resolution. In a third step, the background is cleaned up and the image is converted to binary. Finally, text bounding boxes are refined (repeating steps 2 and 3) by using the extracted items as strokes. The final output produces two binary images for each text box which can then be passed to any standard OCR software.

The system has been tested on images from a wide variety of sources, including newspapers, magazines,, photographs, digitized video frames, etc. Of the 21820 characters and 4406 words in these test images, 95% of the characters and 93% of the words have been successfully extracted by the system. Of these 14703 characters and 2981 words are believed to be OCR-readable fonts, and 84% of the characters and 77% of the words are successfully recognized by a commercial OCR system.

5. Learning in Vision

5.1 New formulation of Control Policies based on Markov Decision Processes and Reinforcement Learning

The original paradigm for SLS was to use decision trees (or other classifiers) to evaluate intermediate data results at each level of representation, and to use some mechanism to choose how to transform data from one level of representation to the next. We developed a model for applying reinforcement learning to object recognition in which each level of representation is viewed as a continuous feature space defined by its measurable attributes, and control policies are learned using a combination of reinforcement learning and neural networks that map points in the feature spaces onto optimal actions. An initial implementation of this new approach was completed in December, 1995. In the first major test of this system, it was 10-for-10 in recognizing rooftops in aerial images of Ft. Hood, TX. These results (which use a reduced library of visual procedures) were reported in (Draper 1996).. Prof. Bruce Draper has joined Colorado State University and will continue developing this work in reinforcement learning in applications related to the Automatic Population of Geospatial Databases (APGD) program.

5.2 Real-time interactive classification

Manual generation of training examples for supervised learning is an expensive process. One way to reduce this cost is to produce training instances interactively that are highly informative. The feasibility of such an approach has been demonstrated on an image pixel classification task that is the front-end to many higher level reasoning applications that can make useful inferences about the contents of the image. However, the construction of pixel classifiers is a labor-intensive task involving user interaction to manually select feature sets, manually select local training data for each desired object class, and then to provide feedback as a result of classification for additional refinement until satisfactory global classification is achieved.

Thus, the prototype system we have implemented is an exploration into a new classification paradigm. We have developed a prototype interactive tool (Piater and Utgoff 1997) that allows the user to immediately see the result of selecting incremental training data so that he can adjust the further selection on the basis of inaccurate classification. This system shows that the incremental classifier converges to satisfactory performance after a very small number of training instances and required only a fraction of the typical human effort to provide them. This suggests an interactive real-time 3D visualization tool for incremental classification of terrain in aerial images. This now allows interactive training of the classifier with the user examining the world data from more natural and understandable viewpoints that show the sensor data in the context of its three-dimensional characteristics, e.g. from a 45 degree downward oblique view, where sides of objects and terrain are more understandable. Classification results can then be rapidly overlaid onto the terrain model using a variety of graphic display techniques, and with incremental real-time updating of training data.

6. Miscellaneous Related Research

6.1 Persistent Data Management for Visual Applications

Visual applications need to represent, manipulate, store, and retrieve both raw and processed visual data. Existing relational and object-oriented database systems fail to offer satisfactory visual data management support because they lack the kinds of representations, storage structures, indices, access methods, and query mechanisms needed for visual data. We have previously argued that *extensible visual object stores* offer feasible and effective means to address the data management needs of visual applications (Draper 1993; Draper). Such a visual object store is under development at the University of Massachusetts for the management of persistent visual information. ISR4 is designed to offer extensive storage and retrieval support for large, complex sets of visual data , customizable buffering and clustering, and spatial and temporal indexing, along with a variety of multi-dimensional access methods and query languages.

6.2 Segmentation of Stroke Lesions in MRI

A collaborative exploratory project with Baystate Medical Center is underway for analysis of stroke lesions in the brain scans (Piater 1996). The goal of this clinical study is the volumetric analysis of damaged cells for people who have suffered an acute ischemic stroke, and their response over time to various forms of treatment involving the lowering of blood pressure in the period immediately following the stroke. This requires the segmentation of brain lesions where there is generally a core of dead tissue (infarct) and a surrounding area of damaged tissue that either might recover or die (penumbra). The change in the size of the lesion over a varying period of time (several days, weeks, and/or months) will be correlated with qualitative assessment of patient functionality and the different forms of treatment

6.3 Weighted Bipartite Matching for 3D Correspondence and Rigid 3-D Motion

A closed form solution has been developed for the problem of determining correspondences between two sets of 3D points for which the number of points in the sets is not the same (Cheng, Wu et al. 1996). This is the general 3D rigid motion problem and the solution is based on a decomposition of the correlation matrix eigenstructure. Using a heuristic measure of point pair affinity derived from the eigenstructure, a weighted bipartite matching algorithm has been developed to determine the correspondences in general cases where missing points occur. The use of the affinity heuristic also leads to a fast outlier removal algorithm, which can be run iteratively to refine the correspondence recovery.

References

Cheng, Y., V. Wu, R. Collins, A. Hanson and E. Riseman. (1996). "Maximum-Weight Bipartite Matching Technique and Its Application in Image Feature Matching," Proc. SPIE Visual Comm. and Image Processing, Orlando, FL.

Collins, R. T., A. R. Hanson, E. M. Riseman, C. O. Jaynes, F. Stolle, X. Wang and Y. Q. Cheng. (1996). "UMass Progress in 3D Building Model Acquisition," Proc. ARPA IUW, Palm Springs, CA, pp. 305-315.

Das, M. and E. Riseman. (1997a). "Feature Selection for Robust Color Image Retrieval," Proc. DARPA IUW, New Orleans, LA.

Draper, B. (1996a). "Learning Grouping Strategies for 2D and 3D Object Recognition," Proc. DARPA IUW, Palm Springs, CA, pp. 1447-1454.

Draper, B., A. R. Hanson and E. M. Riseman. (1993). "ISR3: A Token Database for Integration of Visual Modules," Proc. DARPA IUW, Washington D.C., pp. 1155-1161.

Hoepfner, K., C. Jaynes, E. Riseman, A. Hanson and H. Schultz. (1997). "Site Reconstruction from IFSAR," Proc. DARPA IUW, New Orleans, LA.

Jaynes, C., M. Marengoni, A. Hanson, E. Riseman and H. Schultz. (1997). "Knowledge-Directed Reconstruction from Multiple Aerial Images," Proc. DARPA IUW, New Orleans, LA.

Jaynes, C., E. Riseman and A. Hanson. (1997). "Building Reconstruction from Optical and Range Images," Proc. CVPR, San Juan, Puerto Rico.

Jaynes, C., F. Stolle, H. Schultz, R. Collins, A. Hanson and E. Riseman. (1996). "Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images," Proc. ARPA IUW, Palm Springs, CA, pp. 479-490.

Jensen, F. V. (1996). An introduction to Bayesian networks, New York: Springer Verlag, Inc.

Marengoni, M., B. Draper, A. Hanson and R. Sitaraman. (1996). "Placing Observers to Cover a Polyhedral Terrain in Polynomial Time," Proc. IEEE Workshop on Applications of Computer Vision, Sarasota, Florida.

Piater, J., "The Baystate Stroke Project: Segmentation of Stroke Lesions", TR96-48, Comp. Sci. Dept., Univ. of Mass. (Amherst) 1996.

Piater, J. H. and P. E. Utgoff. (1997). "Interactively Learning Pixel Classification," submitted to International Conf. on Machine Learning.

Quam, L. H. (1984). "Hierarchical Warp Stereo," Proc. DARPA IUW, New Orleans, pp. 149-155.

Ravela, S. and R. Manmatha. (1997). "A Characterization of Visual Appearance Applied to Image Retrieval," Proc. DARPA IUW, New Orleans, LA

Schultz, H. (1994). "Terrain Reconstruction from Oblique Views," Proc. ARPA IUW, Monterey, CA, pp. 1001-1008.

Schultz, H., F. Stolle, X. Wang, E. Riseman and A. Hanson. (1997). "Recent Advances in 3D Reconstruction Techniques from Aerial Images," Proc. DARPA IUW, New Orleans, LA.

Wang, X., R. T. Collins and A. Hanson, "An Orthographic Facet Image Library for Supporting Site Model Refinement and Visualization", TR95-100, Comp. Sci. Dept., Univ. of Massachusetts (Amherst), 1995.

Wang, X., F. Stolle, H. Schultz, E. Riseman and A. Hanson. (1997). "Using Three-Dimensional Features to Improve Terrain Classification," Proc. CVPR, San Juan, Puerto Rico.

Wu, V., R. Manmatha and E. Riseman. (1997). "Automatic Text Detection and Recognition," Proc. DARPA IUW, New Orleans, LA.

IU at the University of Utah: Extraction of Micro-Terrain Features

William B. Thompson and **Thomas C. Henderson**
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
http://www.cs.utah.edu/projects/robot/micro-terrain/

Abstract

The construction of large-scale geospatial databases remains an expensive and time consuming task. We describe an approach for automatically extracting certain types of linear features with a horizontal extent significantly less than the resolution of the base-level terrain data covering the area in which these structures occur. The first phase of this effort considers two classes of such features: ravines and road cuts and fills. Accurate detection and localization of these features is difficult in even high-resolution elevation data. While they are often apparent in aerial imagery, they are easily missed or confused with other features, making reliable detection based on imagery alone problematic. The key to solving this problem is to utilize techniques which combine photogrammetric analysis of the terrain with focused image understanding methods applied to individual images.

1 Overview

Sensor technology, limitations of photogrammetry, storage constraints, and requirements for real-time rendering and analysis all limit the fidelity with which terrain can effectively be represented in a geospatial database. For certain applications, it is critical that these databases include specific *micro-terrain* features with a lateral extent less than the resolution of base-level terrain description. In current generation terrain modeling systems, topographic micro-terrain is seldom present. Man-made micro-terrain, particularly that associated with road cuts, is sometimes included but often does not correctly correspond to the actual terrain.

This project demonstrats how feature extraction methods which combine image understanding with a terrain analysis based on other sources of information can be used to reliably locate micro-terrain features in a way that improves the utility of terrain models used for simulation and synthetic environment applications. Our initial focus is on two types of embankment features which are of particular relevance to DOD:

- Extraction of natural ravines.
- Extraction of road cuts and fills.

Embankments are terrain features that critically affect the realism of ground warfare simulations. They provide concealment while also constituting potential barriers to traversibility. Great difficulty is associated with accurately extracting embankments from source data. Embankments are long, narrow features with a width typically much smaller that the spatial resolution of the terrain data used to construct geospatial databases. While they are often apparent in high resolution aerial imagery, reliable detection is difficult and the visual signature of an embankment is easily confused with other commonly occurring terrain and cultural features. As a result, enhancing the realism of such features in terrain databases currently requires substantial manual processing.

We are addressing these problems with an approach aimed at achieving the following objectives:

- Ability to accommodate a variety of terrain types and covers.

This work was supported by the Defense Advanced Research Projects Agency, contract number pending.

- Tolerance of source data of variable quality and uncertain pedigree.
- Ability to exploit existing tools.
- Easy insertion of results into existing tools.

Our interest lies in cartographic features that are too small to be extracted using stereo photogrammetry and too indistinct to be found reliably using standard image understanding methods alone. Our approach combines existing methods for extracting terrain data and features with standard image processing algorithms in order to extract information not available from either source alone. While micro-features are absent from terrain models produced using conventional photogrammetric means, the presence of such features can often be inferred from even low resolution elevation maps.

Ravines are erosional features generated by large-scale processes, even if the final effect is visible mostly on a fine-scale. As a result, hydrological analysis can be used to predict where such erosion is most likely to occur. Effective algorithms exist for performing this analysis even on low-resolution, error-full representations of the terrain skin.

Except for unmaintained tracks, road construction involves local terrain modifications. When a road crosses a slope transversely, the side-to-side cross-section of the road must be maintained at or near the horizontal. For roads going up or down a steep slope, switchbacks and/or road cuts and fills are often introduced to improve trafficability. 2-D information about road position, obtained from available sources, can be combined with a 3-D analysis of the local terrain to produce predictions about where cuts and fills would have been utilized in the road building process.

As shown in Figure 1, our approach uses predictions about the possible existence and approximate locations of ravine and road cut features to drive a top-down, model-guided image understanding process. The image understanding component confirms the hypothesized presence of terrain features and refines positional estimates, rather than perform a bottom-up extraction of the features themselves. The restricted nature of this task allows the use of simple and reliable image analysis methods that are tolerant of wide variability in input data.

While the intial scope of this project is limited to a set of feature-specific methods, the features themselves are often critical to the utility of the entire geospatial database, particularly when used in support of Synthetic Environment (SE) applications. Existing tools are not able to extract these features without major human effort. The solutions outlined here depend on the use of "context" to direct image understanding methods and to integrate the results of that analysis into a consistent terrain database. Our hope is that the approach will generalize to additional classes of features for which the ways in which the features interact with the surrounding terrain provide enough information to guide targeted IU analyses.

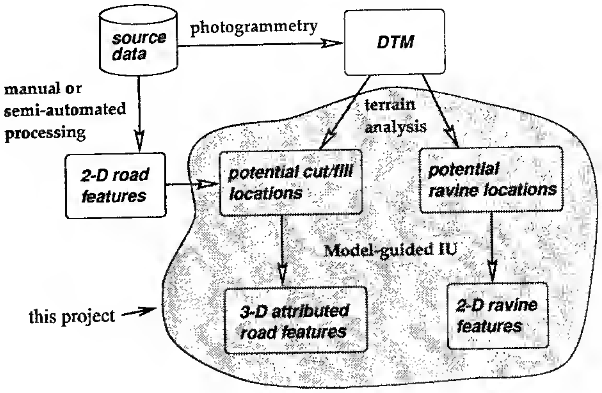

source data
photogrammetry
DTM
manual or semi-automated processing
terrain analysis
2-D road features
potential cut/fill locations
potential ravine locations
Model-guided IU
this project
3-D attributed road features
2-D ravine features

Figure 1: Combining terrain analysis and image understanding for micro-terrain feature extraction.

2 Micro-Features in the Context of Modeling and Simulation (M&S) Applications.

Real-time modeling and simulation applications are increasingly emphasizing realism in both appearance and behavior (Figure 2). The current state-of-the-art in terms of fielded systems is typified by the Army's Close Combat Tactical Trainer (CCTT) [Pope *et al.*, 1995]. While based on real source data, CCTT is a *geotypical* simulation, appropriate for generic training but not for gaining experience with a specific geographic area. As image generators and other computational engines involved in implementing a simulation become more powerful, it becomes increasingly important to be able to produce richer and more accurate models of the relevant 3-D terrain structure and to update these models as changes occur in the world they are representing. Thus, extraction and representation of veridical features is

geotypical	⇒	geospecific
static databases	⇒	changeable databases
texture mapped	⇒	image mapped
training	⇒	mission rehearsal
realistic	⇒	*real*

Figure 2: Trends in real-time simulation.

increasingly essential.

One of the ways that next generation image generators such as the Evans & Sutherland Harmony system achieve improved visual realism is by draping actual aerial imagery over the terrain skin, rather than using geotypical texture mapping. Imagery is becoming part of the primary source data, rather than serving solely as a secondary data source on which photogrammetry and feature extraction is performed. While the visual impact can be dramatic, it is critical that features in the world model such as buildings, roads, and critical topography be represented and localized in a manner consistent with the draped imagery used for rendering. Image Understanding methods are thus likely to play a much more central role in future terrain database creation activities.

It is important that new approaches to the construction of geospatial databases provide improvements in economy -- measured in cost and time -- as well as in the quality of the resulting model. High-resolution IFSAR and high-resolution, highly calibrated photogrammetric stereo is expensive to collect and process and is not likely to be available prior to need for many geographic areas of potential interest to the M&S community. In time-critical situations, the use of terrain feature extraction methods which depend on such data may force significant delays in the model-building process as the needed data is acquired. As a result, it is important to understand the minimum necessary source data quality required to determine relevant information and to build data extraction tools able to compensate for deficiencies in existing source data.

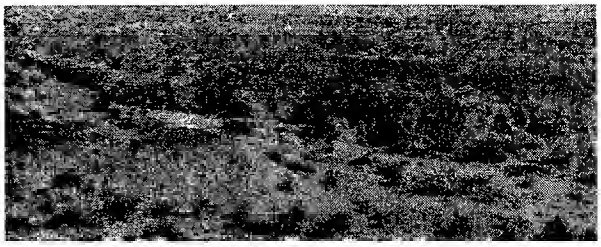


Figure 3: Shallow wash located within Range 400.

3 Ravine Extraction

While ravines and dry washes are usually at least partially visible on aerial photographs, accurate detection and localization is difficult. In addition, ravines are easily confused with roads, tracks, and other structures commonly appearing in non-urban environments. Photogrammetry fails to extract many ravine features because of their restricted depth and small width relative to the resolution with which terrain elevation is extracted. In addition, the photogrammetric overlaps usually used for non-urban terrain preclude the ability to see into many ravine bottoms or measure the slopes of their sides.

The dry wash shown in Figure 3 is located in the live fire range (Range 400) of the USMC Air Ground Combat Center, located at Twentynine Palms, CA. The wash is 1m-2m deep and 2m-3m across. Though *very* small compared to the resolution at which terrain features are usually modeled, such ravines are of critical tactical significance to dismounted infantry, and therefore all units in a combat force.

The nominal resolution of elevation data on which terrain models are based is commonly on the order of 30m or greater. Due to the smoothing inherent in the manner in which the elevation data is obtained, the effective resolution, measured in terms of the size of distinct features apparent in the data, is much coarser. Thus, even with DEM data finer than a 30m grid, terrain structure such as shown in Figure 3 is likely to go unrepresented. Nevertheless, coarse resolution DEMs can be used to predict likely locations where smaller terrain deformations are to be expected.

Hydrological analysis based on digital elevation models is now a standard function in many geographic information systems (GIS). When combined with appropriate resampling and interpolation methods, such operations can be used to effectively

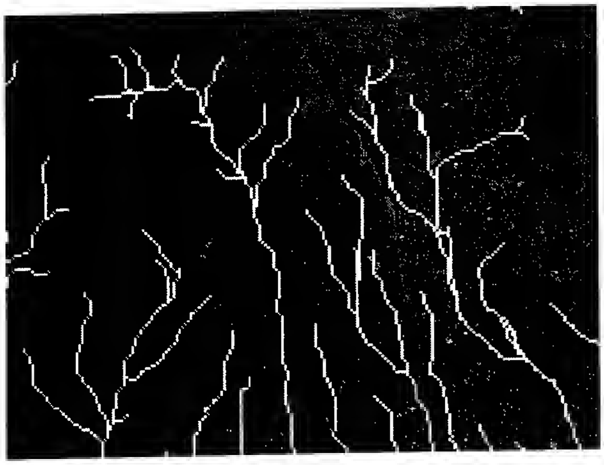


Figure 4: Hydrologic flow analysis based on 30m DEM of Range 400, Overlayed on Orthoimage.

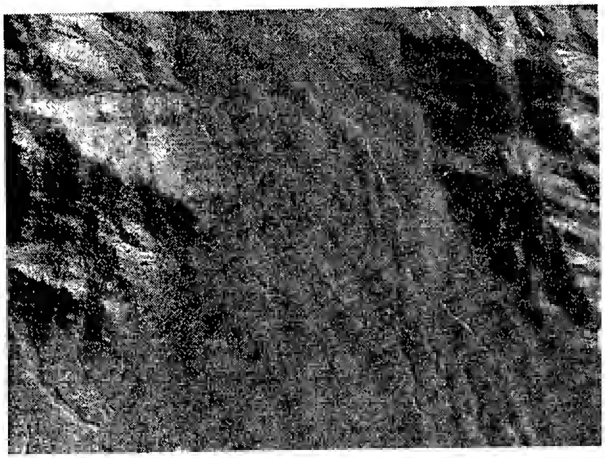


Figure 5: Shaded relief of Range 400 based on high-resolution DEM.

estimate the existence and location of ravine features using DEM data with a post spacing significantly greater than the width of the features of interest. Figure 4 shows the results of applying such a hydrological analysis to a 30m DEM of a portion of Range 400, with the results overlayed on top of a higher-resolution aerial image of the same area.

One of the advantages of using Range 400 as a demonstration area is that high quality DEM data is available with a post spacing of 1m and a nominal precision of 0.1m. For example, Figure 4 can be compared with Figure 5, which shows a shaded relief rendering of the same area, based on the higher resolution DEM. This provides the ability to compare the results of augmenting standard resolution terrain data with image understanding techniques with results based on high precision (and expensive) photogrammetry (e.g., [Richbourg *et al.*, 1995, Richbourg and Olson, 1996, Henderson *et al.*, 1997]).

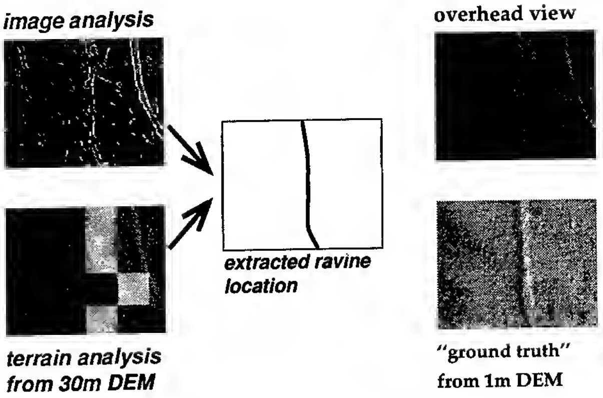


image analysis
overhead view
extracted ravine location
terrain analysis from 30m DEM
"ground truth" from 1m DEM

Figure 6: Combining terrain analysis and IU for ravine extraction.

Figure 6 illustrates our approach. On the upper right of the figure is a section of the Range 400 orthoimage corresponding to a 120m by 150m area on the ground. In the upper left is the output of a Canny edge detector applied to this image. The lower right shows the same area, displayed as a shaded relief rendering of the high resolution DEM data. Essentially, this indicates the "ground truth" topography. The lower left indicates the results of applying automated ravine estimation to a 30m resolution DEM covering the full Range 400 area. The center of the figure illustrates how image and terrain analysis can be combined for ravine extraction (see [Thoenen and Thompson, 1997]).

4 Road Cut and Fill Extraction

For a simulation to accurately reflect tactical reality, it is important to be able to reliably determine whether or not road cuts and fills need to be added to road segments in a geospatial database. Figure 7 shows a view generated from the U.S. Army Close Combat Tactical Trainer (CCTT) Central U.S. database (CCTT Primary One). The road cut, which appears as a long trench cut through a hillside, was generated by an automatic cut and fill insertion tool that first drapes the road onto the terrain skin and then inserts cuts and fills whenever needed to keep the road grade from exceeding a preset steepness.

Figure 7: Incorrect road cut appearing in CCTT Primary One due to incorrect automatic insertion tools.

It is one of a large number of road cuts and fills in CCTT Primary One that are obviously incorrect to anyone viewing the database. While improved cut and fill insertion techniques might reduce the incidence of implausible configurations such as shown here, there is no way to accurately represent what is actually on the ground without revisiting the source data.

In most geospatial database construction systems, information about the terrain skin and the location of roads is independently determined and then represented in separate layers in a GIS. A merging operation must then be performed to create a terrain model that is realistic and behaves in a manner consistent with the desired semantics of the simulation. Road networks are almost always initially extracted as 2-D features, with no information explicitly available about superelevation or grade. Three general approaches are possible for adding this information to produce a full 3-D representation of the road surface (Figure 8): The road can be "draped" over the terrain, adapting to the terrain surface; the terrain can be locally deformed so as to make the road surface plausible and to blend the road and terrain in a natural manner; or explicit cuts and fills can be introduced to deal with discrepancies between the desired geometry of the road surface and the shape of the underlying terrain.

As simulations move from geotypical to geospecific databases, it becomes increasingly important to determine not just where road cuts and fill are plausible, but where they actually occur. Our proposed method for detecting road cuts and fills will proceed as follows:

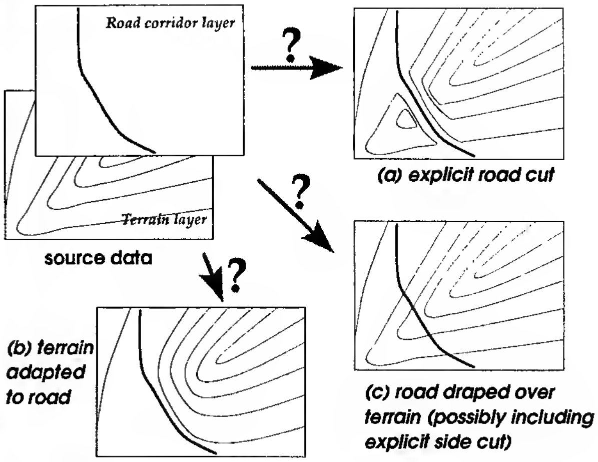

Road corridor layer
Terrain layer
source data
?
?
?
(a) explicit road cut
(b) terrain adapted to road
(c) road draped over terrain (possibly including explicit side cut)

Figure 8: Road cut insertion requires merging 2-D road locations with the 3-D terrain skin by (a) introducing an explicit road cut, (b) modifying the terrain skin to accommodate the road, or (c) draping the road over the terrain.

- Search in the imagery for roads present in the road corridor layer of the source data GIS. This step is similar to the image analysis operations used for ravine extraction.
- Use civil engineering principles applied to road corridor data draped over the base-level terrain skin to predict where cuts and fills are likely to occur.
- Search along the sides of roads in located in the imagery, looking for textural variations that correspond to the pattern of predicted cuts and fills. The key is to search for patterns of variability, since the actual visual appearance of cuts and fills is impossible to predict.

5 Integration With Existing Tools

To be cost effective, methods developed as part of this project must be easily integrated with other aspects of the geospatial database creation process. Micro-feature extraction methods should be able to utilize the full range of existing geospatial database tools. The results of these micro-feature extraction methods should be easily used by the full range of existing geospatial database tools.

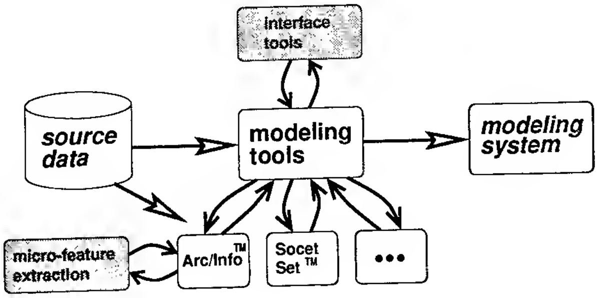


Figure 9: Integrating micro-feature extraction into established procedures for building terrain models.

Geospatial database tools use a much broader range of data formats than do most "pure" image understanding systems. Imagery, elevation data and other specifications of terrain skin, linear features, and 3-D structures must all be represented in geospecific coordinate systems. In the initial phase of this project, all code we develop will be interfaced to the Arc/Info™ GIS software. While Arc/Info is a proprietary system, it is likely to be used by anyone building production-grade geospatial databases.

Figure 9 shows how programs we develop fit into the larger context of geospatial database production. Modeling tools in common usage take in source data; organize, analyze, and process it with the help of tools such as Arc/Info™ and GDE's Socet Set™; and create models using a language appropriate to the simulation or mission planning systems required by the end user. By coupling our system to Arc/Info, we facilitate compatibility with any model generation system that uses Arc/Info as a support tool.

6 Evaluation

Too often, image understanding methods are evaluated in isolation instead of in the context of the larger systems that they are typically embedded within. The IU methods described in this paper should be viewed as adding value to existing geospatial database construction processes. For evaluation purposes, we will start with state-of-the-art simulation databases, which often have been constructed at substantial expense. These databases will have known deficiencies which affect critical aspects of the realism of simulations. The end result of the methods we have described above will be to improve these existing terrain databases. Thus, the appropriate evaluation process involves comparing the databases before and after the processing we propose. This allows for an operational assessment by letting end users interact with simulations involving the original and improved terrain databases.

Given believable ground truth data, quantitative measures can also be developed. In the case of Range 400, the availability of 1m DEM data will aid this process. Quantitative evaluation will be based on how well IU methods combined with lower-resolution DEMs can approximate the ravine features apparent in the higher-resolution DEMs. Richbourg's work in developing computational techniques for analyzing concealment features in the Range 400 data provides a valuable starting point [Richbourg *et al.*, 1995, Richbourg and Olson, 1996]. The NIST-TEC-LADS project investigating landform extraction from the Range 400 data is also relevant.

References

[Henderson *et al.*, 1997] T. C. Henderson, S. Morris, and C. Sanders. Ridge and ravine detection in digital images. In *Proc. DARPA Image Understanding Workshop*, May 1997.

[Pope *et al.*, 1995] C. N. Pope, M. Vuong, R. G. Moore, and S. S. Cowser. A whole new CCTT world. *Military Simulation and Training*, (5), 1995.

[Richbourg and Olson, 1996] Robert Richbourg and Warren K. Olson. A hybrid expert system that combines technologies to address the problem of military terrain analysis. In *Expert Systems and Applications*, 1996.

[Richbourg *et al.*, 1995] R. Richbourg, C. Ray, and L. L. Campbell. Terrain analysis from visibility metrics. In *Conference on Integrating photogrammetric techniques with scene analysis and machine vision II, (SPIE Proceeding Volume 2486-2)*, Orlando, FL, April 1995.

[Thoenen and Thompson, 1997] G. W. Thoenen and W. B. Thompson. Extraction of micro-terrain ravines using image understanding constrained by topographic context. In *Proc. DARPA Image Understanding Workshop*, May 1997.

Image Understanding Research at Colorado State University *

Bruce A. Draper J. Ross Beveridge
Computer Science Department
Colorado State University
Fort Collins, CO 80523
draper/ross@cs.colostate.edu

Abstract

Colorado State University is initiating a new project on learning control strategies for object recognition. It is our belief that the current library of IU algorithms is sufficient for solving many practical tasks if we can only learn to sequence them properly. We are investigating the use of open-loop and closed-loop control policies for sequencing IU algorithms, emphasizing the use of Markov decision models and reinforcement learning to derive closed-loop object recognition policies. This work is being conducted in the context of the Automatic Population of Geospatial Databases (APGD) project, where it will be used to learn object recognition strategies for finding buildings, roads and other objects of interest in aerial images.

1 Introduction

Image understanding (IU) research at Colorado State University (CSU) is based on two simple premises:

- IU researchers have made tremendous progress in developing algorithms for many aspects of the computer vision problem, including feature extraction, shape reconstruction and model matching.
- Despite this progress, there are very few practical IU systems because they are too difficult to build and too brittle with respect to changes in the domain or task statement.

Based on these observations, researchers at CSU are developing recognition strategies for the DARPA Automatic Population of Geospatial Databases (APGD) program based on a new approach. Instead of hand developing new algorithms for specific IU tasks, we approach IU as a control problem. Our hypothesis is that many practical IU problems can be solved through existing techniques, provided we can learn to select the proper sequence of algorithms. Our goal is to develop the technology for learning to recognize objects from examples by training control strategies that select sequences of IU algorithms based on the object to be recognized and the domain context.

2 Background

Over the last twenty years, computer vision researchers have divided the field into ten or twenty (or more) topics, each with a narrowly-defined problem focus. Within these subfields, theories have been developed and specific algorithms have been proposed. As a result, there are now good and improving algorithms for feature extraction (including points, edges, lines, curves and regions), stereo analysis, multi-frame feature tracking, depth from motion (two-frame and multi-frame), shape matching, color matching, pixel matching (a.k.a. appearance matching), and 3D pose determination, and new algo-

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under contract 96-14-201 monitored by the Army Topographic Engineering Laboratory (TEC), and contracts DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office

rithms are being developed as this is written.

At the same time, other researchers (particularly within the DARPA community) have concentrated on building end-to-end systems that solve specific IU tasks. For example, in recent years competing systems have been developed for recognizing and reconstructing buildings from aerial images [McGlone & Shufelt, 1993; Lin *et al.*, 1994; Jaynes *et al.*, 1996], while other systems have been built for recognizing military targets, road networks and terrain features. These systems can be viewed as the intellectual descendents of earlier knowledge-based systems that exploited knowledge about objects and domains to create special-purpose object recognition strategies (e.g. [McKeown *et al.*, 1985; Hwang *et al.*, 1986; Draper *et al.*, 1989]).

If we take a close look at the recent work on building reconstruction [McGlone & Shufelt, 1993; Lin *et al.*, 1994; Jaynes *et al.*, 1996], we can draw several conclusions:

- These systems were built by sequencing standard IU algorithms for line extraction, line grouping, shadow analysis and graph matching/traversal. (Some of these systems refined previous algorithms in these areas.)
- These systems are special-purpose to the extent that they reconstruct a single type of object within a specific domain. They will not work if the target object class or domain is changed significantly.
- It is difficult to analyze these systems. There is no underlying theory by which to gauge their performance, nor is there an analytical method for predicting at what point they will break. It is difficult even to compare them, since each system makes slightly different assumptions about the imagery.
- These systems are difficult to build. Each system took months or even years of highly skilled labor to construct, and even so most of these systems could be improved given more time.

Our project is an effort to generalize from earlier object recognition efforts. Like these previous systems, our goal is to sequence existing IU algorithms in order to recognize specific objects within limited contexts. Unlike previous efforts, we will neither construct these algorithm sequences by hand nor build a "knowledge base" with rules for selecting algorithms (such knowledge bases have proven error-prone and difficult to build in the past; see [Draper & Hanson, 1991]). Instead, we will model IU as a control problem, in which the goal is to select the best sequence of algorithms for recognizing a given object class. More specifically, we model the IU control problem as a Markov decision process, in which the goal is to train a control policy that selects algorithms so as to maximize the expected reward over time, where the reward is a weighted function of cost versus accuracy. Our aim is a system that is general-purpose and robust in the sense that it can be retrained to recognize a wide variety of objects in various domains, and theoretically sound in the sense that it will converge to the optimal control policy as the training set size increases.

3 Relevance to APGD

To the extent that we are successful, our technology for learning object recognition strategies could be used within the APGD context for context-based algorithm control. We imagine a system in which image analysts begin to populate a database by outlining and labeling objects of interest in images. As the analysts work, the system uses these labeled object instances as training samples for learning object recognition policies. When the strategy for an object class is trained, the system will take over for the analyst and automatically label any remaining instances of the object class and enter these instances into the geospatial database. Moreover, as new images are acquired the system will automatically label objects instances and update the database.

Through continued training, a system which learns strategies could in principle adapt to recognize any visually distinct object class. Moreover, because the system will be based on a Markov decision process model and reinforcement learning, there are analytical reasons to believe that the control policies it learns will be sound; this is not true of current hand-crafted systems. Finally, because the reinforcement learning process makes a series of predictions about the intermediate data it generates, the system should be able to detect when its control policies are failing (perhaps because of a new variant of an object class or because of a change in the domain) and to ask the image analyst for new training samples, rather than make possibly disastrous mistakes.

4 Image Understanding as a Control Problem

Underlying this work is the notion of IU as a control problem. In casting IU as a control problem, we make the assumption that there is a library of available IU algorithms (such as the ones mentioned in Section 2), and that these algorithms are sufficient to solve many interesting IU problems. We also assume that the system is given an IU task of the form "Find the X in Y", where X is an object to be recognized and Y is a set of images; examples of X in the APGD context include buildings, roads, trees and power lines, while an example of Y might be EO nadir-view images of a constrained geographic area such as Bosnia or Somalia. The challenge for the control system is to find a sequence of algorithms (ideally, the best sequence of algorithms) for finding the target object in the given domain.

Readers should note that we propose to control algorithms, not physical devices such as cameras or platforms. Other researchers have addressed physical camera control in the context of active vision research. Also, there is a harder version of the IU problem in which the object being searched for is unspecified (i.e. "What's in the image, and where is it?"). This form of the problem includes the object indexing problem, which is outside of the scope of this work.

4.1 Open-. vs. Closed-loop Control

Once the decision is made to cast IU as a control problem, several interesting questions emerge. The first of these questions is whether the system is an open-loop or closed-loop control system. In open-loop control, the system selects a fixed sequence of actions for each task. Open-loop control systems have the advantage that recognition strategies can be easily expressed as sequences of algorithms, and open-loop policies can be learned by searching the space of algorithm sequences; Brown & Roberts [1994], for example, use genetic algorithms to search for the best sequence of algorithms for a specific automatic target recognition (ATR) task.

On the other hand, open-loop control systems have the disadvantage that they are unable to adjust to unexpected events. For example, one strategy for finding buildings in aerial images might be to first extract building corners, where the camera viewpoint determines the expected image angle. Unfortunately, if the corner detector fails in an open-loop system (perhaps because of an error in the estimated viewpoint) the open-loop strategy will continue to apply the remaining algorithms in the sequence, even though the data produced by the first step was erroneous. Closed-loop systems, on the other hand, do not produce explicit sequences of actions. Instead, they select an algorithm at each stage of the process based on the results of the previous processing step. This gives closed-loop systems the ability to react to unexpected events during processing, for example by backtracking and selecting another algorithm.

More formally, closed-loop control strategies are defined by *policies* that map states of the system onto actions (i.e. algorithms), where system states are determined by measuring feature attributes. Closed-loop control is more powerful than open-loop control, and it is our belief that variations among images within a domain and the inherent unreliability of many IU algorithms imply that closed-loop policies will be needed for robust control. This is still a hypothesis, however, and one of our tasks in this project will be to compare closed-loop strategies learned through reinforcement learning with open-loop strategies learned through search.

4.2 Expert Knowledge vs. Machine Learning

Another question is whether control decisions come from reasoning about expert knowledge or whether they are learned automatically from experience. We believe that although many variations on expert systems have been tried for IU (e.g. SPAM [McKeown *et al.*, 1985], SIGMA [Hwang *et al.*, 1986], PSEIKI [Andress & Kak, 1988], the Schema System [Draper *et al.*, 1989] and more recently OCAPI [Clement & Thonnat, 1993]), they have always proven to be difficult to build and harder to extend. Moreover, even when they produce acceptable results on a limited set of images, it is difficult to tell whether the control system has performed well or not. We base our conclusions in part on our own past work. In [Draper & Hanson, 1991] we used the Schema System to illustrate problems inherent in the hand-construction of expert knowledge bases for IU, while in [Draper *et al.*, 1996] we give a broader scope to these issues.

The alternative is to build a system that learns control strategies based on examples. Brown & Roberts [1994] is one example of a system that learns open-loop control policies bases on examples ([Chen & Mulgaonkar, 1992] is another). We propose to build on our earlier work [Draper, 1996a; Draper, 1996b] by building a system that uses reinforcement learning to automatically acquire closed-loop control policies from examples. (In related but different work, [Peng & Bhanu, 1996] used reinforcement learning to select parameters for IU algorithms.).

It is our contention that in the long run machine learning is the best source of control strategies. A fielded APGD system, for example, must be able to adapt to new object classes in new domains. A system requiring expert modification and recertification for each new object or domain is not acceptable to the military (or indeed to any user). Instead, it is our goal to show that robust closed-loop control policies for object recognition can be learned by observing an expert.

4.3 General-purpose vs. Object-specific Attributes

A critical issue for closed-loop control systems is generating object-specification attribute measures for intermediate data representations. In a closed-loop system, a control policy selects actions (i.e. algorithms) based on the current state of the system. The system state in turn is a reflection of attributes that can be measured for the features that have been extracted up to that point. For example, in the hypothetical closed-loop control policy for recognizing buildings mentioned in Section 4.1, the first action was to extract corners from the image data. The second action was then selected based on the number and quality of corners found in the first step.

Clearly, closed-loop control policies can only outperform open-loop action sequences if the attributes of the image features provide meaningful feedback. In earlier experiments on learning to recognize buildings we provided a system with routines for computing sophisticated feature attributes, including an algorithm which measured how much of a shadow a feature cast (based on the known camera viewpoint). This attribute proved to be critical; as reported in [Draper, 1996a] (page 1453), the number of false alarms detected by the trained system dropped significantly when the shadow attribute was introduced.

In this project, we intend to have the system develop meaningful attributes on its own. Some of these attributes will be learned, while others will be deduced from *a priori* models. Attributes may be derived from many levels of representation, including image properties, such as color and texture, and object geometry. For instance, we will extend our earlier work with linear machine decision trees to learn the apparent color of objects in outdoor imagery [Buluswar & Draper, 1994] to train attributes that match image features to the expected textures (and if available, colors) of objects. We will also build on current work for matching features derived from geometric object models. Prior examples include our past work on matching horizons derived from terrain maps to imagery [Beveridge & Balasubramanian, 1997] and our multisensor system for predicting observable features of CAD vehicle models [Stevens & Beveridge, 1997]. The specific feature sets developed in these examples do not carry over directly to the APGD domain. However, the principles employed by the algorithms generating these features are applicable to APGD. We will also be expanding our work on a set of algorithms for learning probe sets and/or eigenvalue measures from example features extracted from images [Stevens *et al.*, 1997].

5 Evaluating Recognition Strategies

In order to evaluate our ability to learn closed-loop object recognition policies, we will apply our system to the APGD Fort Hood dataset[1] and test its ability to recognize objects of strategic interest. In particular, we will begin by training recognition policies to find buildings and roads. Then we will test how easily the system can adapt to new tasks by training it to recognize two more object classes, to be determined jointly by ourselves and representatives of the Army Topographic Engineering Center (TEC). Finally, we will adopt a new dataset from a different domain to see how easily the system adapts from one setting to another.

In general, when evaluating our system, the quality of a control policy will be measured by a utility function that balances accuracy and cost. (The relative weight of accuracy vs. cost is determined by the user prior to training.) To measure the effectiveness of the learning system, however, we must seperate the performance of the control policy from the quality of the underlying IU algorithms. To do this, we will compare control policies against two standards. The first standard is the result of an exhaustive search of the space of open-loop strategies. (We can compute this because the space of open-loop strategies is much smaller than the space of closed-loop policies.) Closed-loop policies trained through reinforcement learning will then be compared to the optimal open-loop strategy according to the user-defined utility function. Second, we will compare closed-loop policies to each other. Although there is no way to know what the true optimal closed-loop policy for a given task may be, if we train multiple closed-loop policies we can compare them to each other, determining which are the best (and by how much).

6 Practicum: Khoros and the IUE

At a more mundane level, work on IU as a control problem can only proceed if libraries of IU algorithms are accessible. One of the goals of the Image Understanding Environment (IUE) [Mundy *et al.*, 1992] is to disseminate libraries of IU "task" objects, which are implementations of IU algorithms. To this end, we have been actively contributing to the IUE effort. Our most recent contribution is a target detection algorithm for use in IR imagery. This algorithm was selected as an archetypical ATR algorithm and it is based loosely on the concepts of a sliding window detector set out by Nguyen [Nguyen, 1990]; it is of practical interest because it is used as the first phase of a two phase target detection algorithm on the Unmanned Ground Vehicle Program's Semi-Autonomous Scout Vehicles. We are also developing a version of our optimal line segment matching system for release with the IUE.

At the same time, we are forced to recognize the limited state of the current IUE task library. We will therefore be developing our learning system within the Khoros image processing environment [Rasure & Kubica, 1994] which at the moment has a more extensive library of (mostly low-level) computer vision algorithms, and we will be extending this library as necessary. Fortunately, long-term plans call for the IUE to become compatible with Khoros, and it is our hope to be able to access both the IUE and Khoros task libraries in the relatively near future.

7 Recent Accomplishments

Although the main thrust of this paper is to look forward to our project on learning object recognition policies, we thought it would be useful to briefly summarize some of our recent accomplishments, and in particular to expose some underlying intellectual connections between our previous work and our new project.

7.1 Automatic Target Recognition

Over the past three years we have devoted considerable attention to the development of

[1] For readers unfamiliar with the Fort Hood data, it is a collection of approximately twenty high-resolution black-and-white aerial images of Fort Hood, TX, including approximate camera parameters for each image.

Table 1: Confusion matrix for Multisensor Target Identification. Correct identification rate is 27/35 (77%). The two entries marked with '*' are cases where hypothesis generation failed to suggest the correct target type.

		Multisensor System ID			
		M113	M901	M60	Pickup
True Target ID	M113	7		1	1
	M901	1^*	5	2	1^*
	M60		1	7	1
	Pickup				8

new model-based ATR techniques for multisensor imagery. This work was supported by the DARPA IU Program as part of the Unmanned Ground Vehicle (UGV) program's RSTA (Reconnaissance, Surveillance and Target Acquisition) activity. A detailed report on this effort appears in [Beveridge *et al.*, 1997a]; Here we discuss only those portions of the work that are relevant to the APGD project.

The CSU ATR system was a three-stage target detection and recognition system that performed well on the difficult Ft. Carson data set [Beveridge *et al.*, 1994]. On 35 target identification tasks involving 4 targets, the CSU system correctly identified 27 out of 35 (77%) of the targets. If we neglect difficult cases, such as distant and occluded targets, the correct identification rate is over 90%. The confusion matrix summarizing this result is presented in Table 1.

As a point of comparison, the group from MIT Lincoln Laboratory has used the Ft. Carson dataset in part of the evaluation of their own range-based ATR system [Verly & Lacoss, 1997]. Based upon their performance modeling work, they conclude their approach is only applicable to four of the highest resolution Ft. Carson range images.

Although some aspects of the CSU ATR system are tailored to ATR, some of the technology is applicable to the APGD task. The first stage of the ATR system used a linear machine decision tree to learn the range of apparent colors exhibited by an object under outdoor lighting conditions [Buluswar & Draper, 1994]. Although the Ft. Hood dataset used for the current APGD work does not include color data, this same learning technique can be used to learn combinations of texture measures extracted from black and white imagery. How well an image feature matches the expected texture of an object then becomes an attribute that a closed-loop control system can use for feedback.

The second stage of the CSU ATR system used probing techniques to suggest possible targets and target orientations. The probing techniques derived probe sets from BRL/CAD models of possible targets, and we developed new techniques based on neural networks for efficiently selecting the most relevant probesets [Stevens *et al.*, 1997]. Once again, although the objects being searched for in the APGD domain will be different, probing techniques such as these can be used to develop object-specific feature attributes whenever either object models or substantial training imagery are available. We have also begun to explore the intellectual connections between probing and eigenspace analysis [Nayar *et al.*, 1996; Kirby & Sirovich, 1990; Turk & Pentland, 1991], and are looking for ways to train both probe sets and eigenspace representations from the APGD training samples.

Finally, the third stage of the CSU ATR system performed the final target identification and target pose determination by matching 3D target models to the multisensor image data. By exploiting an iterative predict-and-match cycle between the 3D object model and the multisensor image data, we have demonstrated what we consider to be several significant advances in the state-of-the-art for ground-based multisensor ATR. We have demonstrated an ability to take a rough target pose estimate, i.e. off by as much as 30°, and generate a more reliable estimate accurate to within about 5° [Stevens & Beveridge, 1997]. Further, this is done in the presence of errors in the initial registration mappings between sensors: our algorithm refines sensor registration and 3D target pose as part of the matching process.

Although this work might at first seem unrelated to the current APGD effort, this work implies that our learning systems will have access to state-of-the-art algorithms for matching geometric models to data. Moreover, it demonstrates an ability to propagate evidence for terrain occlusion between sensors and accordingly modify range, IR and color target features during the matching cycle. This means that the CSU ATR system does not try to find features which it can infer are occluded by foreground terrain. Such an ability to reason about why certain features might not be seen will be critical to control systems that must distinguish between features that are missing (or may have changed), and those which simply cannot be seen from the current viewpoint.

8 Conclusion

The current focus of IU research at Colorado State University is on understanding the implications of modeling IU as a control problem, and on building practical systems that learn object recognition strategies from examples. This work is being conducted in the context of the Automatic Population of Geospatial Databases (APGD) project, where it will be used to learn object recognition strategies for finding buildings, roads and other objects of interest in aerial images.

References

[Andress & Kak, 1988] Andress, K.M. and Kak, A.C. "Evidence Accumulation and Flow of Control in a Hierarchical Reasoning System," *AI Magazine* 9(2):75–94 (1988).

[Beveridge & Balasubramanian, 1997] R. Beveridge and K. Balasubramanian. "Camera Orientation Refinement by Matching of Terrain Map Horizons to Ground-looking Imagery," IUW 1997..

[Beveridge *et al.*, 1994] R. Beveridge, D. Panda, and T. Yachik. November 1993 Fort Carson RSTA Data Collection Final Report. Technical Report CSS-94-118, Colorado State University, Fort Collins, CO, January 1994.

[Beveridge *et al.*, 1996] R. Beveridge, C. Graves and C. Lesher. Local Search as a Tool for Horizon Line Matching. In *Proceedings: Image Understanding Workshop*, pages 683 – 686, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[Beveridge *et al.*, 1997a] R. Beveridge, B. Draper, M. Stevens, K. Siejko and A. Hanson. A Coregistration Approach to Multisensor Target Recognition with Extensions to Exploit Digital Elevation Map Data. In Oscar Firschein, editor, *Reconnaisance, Surveilance, and Target Acquisition for the Unmanned Ground Vehicle*, page (to appear). Morgan Kaufmann, 1997.

[Beveridge *et al.*, 1997b] R. Beveridge, C. Graves and J. Steinborn. Comparing Random-Starts Local Search with Key-Feature matching. In *Proc. 1997 International Joint Conference on Artificial Intelligence*, page (submitted), August 1997.

[Bienstock *et al.*, 1990] E. Bienstock, D. Geman, S. Geman, and D. McClure. Development of laser radar atr algorithms: Phase II - Military Objects. Technical report, Mathematical Technologies Inc., Providence, Rhode Island, October 1990. Prepared under Harry Diamond Laboratories Contract No. DAAL02-89-C-0081.

[Brown & Roberts, 1994] C. Brown and B. Roberts. "Adaptive Configuration and Control in an ATR System," *DARPA Image Understanding Workshop,* Monterey, CA. 1994, pp. 467-480.

[Buluswar & Draper, 1994] S. Buluswar and B. Draper. "Non-parametric Classification of Pixels of Varying Outdoor Illumination," *DARPA Image Understanding Workshop,* Monterey, CA. 1994, pp. 727-732.

[Chen & Mulgaonkar, 1992] C. Chen and P. Mulgaonkar. "Automatic Vision Programming," *CGVIP-IU,* 55(2):170-183, 1992.

[Clement & Thonnat, 1993] V. Clement and M. Thonnat. "A Knowledge-based Approach to Integration of Image Processing Procedures," *CGVIP-IU,* 57:166-184, 1993.

[Delanoy *et al.*, 1993] R. Delanoy, J. Verly and D. Dudgeon. Machine Intelligent Automatic Recognition of Critical Mobile Targets in Laser Radar Imagery. *The Lincoln Laboratory Journal,* 6(1):161–186, Spring 1993.

[Draper *et al.*, 1989] B. Draper, R. Collins, J. Brolio, A. Hanson and E. Riseman. "The Schema System," *International Journal of Computer Vision,* 2:209-250, 1989.

[Draper & Hanson, 1991] B. Draper and A. Hanson. "An Example of Learning in Knowledge-directed Vision," *Scandinavian Conference on Image Analysis,* Aalborg, DK, 1991. pp. 189-201.

[Draper, 1996a] B. Draper. "Learning Grouping Strategies for 2D and 3D Object Recognition," *DARPA Image Understanding Workshop,* Palm SPrings, CA, 1996. pp. 1447-1454.

[Draper, 1996b] B. Draper. "Object Recognition as a Markov Decision Process," *International Conference on Pattern Recognition,* Vienna, 1996. pp. 25-29, Vol 4.

[Draper *et al.*, 1996] B. Draper, A. Hanson and E. Riseman. "Knowledge-directed Vision: Control, Learning and Integration," *Proceedings of the IEEE*, 84(11):1625-1637, 1996.

[Gillberg *et al.*, 1990] J. Gillberg, R. Johnston, K. Siejko, and J. Lee. Laser Radar ATR Algorithms. Technical Report DAABO7-87-C-F109, Honeywell Systems and Research Center, Minnesota, May 1990.

[Goldberg *et al.*, 1993] D. Goldberg, K. Deb, H. Kargupta, and G. Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In Stephanie Forrest, editor, *Proc. 5th International Conference on Genetic Algorithms*, pages 56–64. Morgan-Kaufmann, 1993.

[Huttenlocher & Ullman, 1990] D. Huttenlocher and S. Ullman. Recognizing Solid Objects by Alignment with an Image. *International Journal of Computer Vision*, 5(2):195 – 212, November 1990.

[Hwang *et al.*, 1986] Hwang, V.S-S., Davis, L.S., and Matsuyama, T. "Hypothesis Integration in Image Understanding Systems," *Computer Vision, Graphics, and Image Processing,* 36:321–371 (1986).

[Jaynes *et al.*, 1996] C. Jaynes, F. Stolle, H. Schultz, R. Collins, A. Hanson and E. Riseman. Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images. *DARPA Image Understanding Workshop*, Palm Springs, CA 1997, pp. 479-480.

[Kirby & Sirovich, 1990] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103 – 107, January 1990.

[Koenderink & van Doorn, 1979] J. J. Koenderink and A. J. van Doorn. The Internal Representation of Shape with Respect to Vision. *Biological Cybernetics*, 32:211–216, 1979.

[Lin *et al.*, 1994] C. Lin, A. Huertas and R. Nevatia. Detection of Buildings using Perceptual Groupings and Shadows, *International Conference on Computer Vision and Pattern Recognition*, 1994, pp. 62-69.

[McGlone & Shufelt, 1993] C. McGlone and J. Shufelt. "Incorporating Vanishing Point Geometry into a Building Extraction System," *DARPA Image Understanding Workshop,* Washington D.C., 1993, pp. 437-448.

[McKeown *et al.*, 1985] McKeown, D.M. Jr., Harvey, W.A., and McDermott, J. "Rule-Based Interpretation of Aerial Imagery," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 7(5):570–585 (1985).

[Mundy *et al.*, 1992] J. Mundy et al. "The Image Understanding Environments Program," *International Conference on Computer Vision and Pattern Recognition,* San Mateo, CA, 1992, pp. 185-214.

[Nayar *et al.*, 1996] S. Nayar, S. Nene and H. Murase. Real-Time 100 Object Recognition System. In *Proceedings of ARPA Image Understanding Workshop*. Morgan Kaufmann, 1996. http://www.cs.columbia.edu/CAVE/rt-sensors-systems.html.

[Nguyen, 1990] D. Nguyen. An iterative Technique for Target Detection and Segmentation in IR Imaging Systems. Technical Report November, (CECOM) Center for Night Vision and Electro-Optics, 1990.

[Olson, 1994] C. Olson. Time and space efficient pose clustering. In *CVPR94*, pages 251–258, 1994.

[Peng & Bhanu, 1996] J. Peng and B. Bhanu. "Closed-loop Object Recognition using Reinforcement Learning," *International Conference on Computer Vision and Pattern Recognition,* 1996, pp. 538-543.

[Rasure & Kubica, 1994] J. Rasure and S. Kubica. "The Khoros Application Development Environment," in *Experimental Environments for Computer Vision and Image Processing,* H. Christensen & J. Crowley (eds.), World Scientific Press, 1994.

[Rimey, 1995] R. Rimey. RSTA Sept94 Data Collection Final Report. Technical report, Martin Marietta Astronautics, Denver, CO, January 1995.

[Stevens & Beveridge, 1997] M. Stevens and R. Beveridge. Precise Matching of 3-D Target Models to Multisensor Data. *IEEE Transactions on Image Processing,* 6(1):126–142, January 1997.

[Stevens *et al.*, 1997] M. Stevens, C. Anderson and R. Beveridge. Efficient Indexing for Object Recognition Using Large Networks. In *Proc. 1997 IEEE International Conference on Neural Networks* , page (to appear), June 1997.

[Turk & Pentland, 1991] M. Turk and A. Pentland. Face Recognition Using Eigenfaces. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 – 591, June 1991.

[Verly & Lacoss, 1997] J. Verly and R. Lacoss. Automatic Target Recognition for LADAR imagery Using Functional Templates Derived From 3-D CAD Models. In Oscar Firschein, editor, *Reconnaissance, Surveillance, and Target Acquisition (RSTA) for the Unmanned Ground Vehicle.* Morgan Kaufmann, 1997.

Learning to Detect Rooftops in Aerial Images[*]

Marcus A. Maloof[†] Pat Langley[†] Stephanie Sage[†] Thomas O. Binford[‡]

[†]Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306
{maloof, langley, sage}@isle.org • http://www.isle.org

[‡]Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, CA 94305
binford@cs.stanford.edu • http://robotics.stanford.edu

Abstract

In this paper, we examine the use of machine learning to improve the robustness of systems for image analysis on the task of roof detection. We review the problem of analyzing aerial photographs, and describe an existing vision system that attempts to automate the identification of buildings in aerial images. After this, we briefly review several well-known learning algorithms that represent a wide variety of inductive biases. We report three experiments designed to illuminate facets of applying machine learning methods to the image analysis task; one experiment focuses on within-image learning, another deals with the cost of different errors, and a third addresses between-image learning. Experimental results demonstrate that machine-learned classifiers meet or exceed the accuracy of handcrafted solutions and that useful generalization occurs when training and testing on data derived from different images.

1 Introduction

The number of images available to image analysts is growing rapidly, and will soon outpace their ability to process them. Computational aids will be required to filter this flood of images and focus the analyst's attention on interesting events, but current image understanding systems are not yet robust enough to support this process. Successful image understanding relies on knowledge, and despite theoretical progress, implemented vision systems still rely on heuristic methods that remain fragile. Handcrafted knowledge about when and how to use particular vision operations can give acceptable results on some images but not others.

In this paper we explore the use of machine learning as a means for improving knowledge used in the vision process, and thus for producing more robust software. Recent applications of machine learning in business and industry [Langley and Simon, 1995] hold useful lessons to its application in image analysis. A key idea in applied machine learning involves building an *advisory* system that recommends actions but gives final control to a human user, with each decision generating a training case, gathered in an unobtrusive way, for use in learning. This setting for knowledge acquisition is similar to the scenario in which an image analyst interacts with a vision system, finding some system analyses acceptable and others uninteresting or in error. The aim of our research program is to embed machine learning into this interactive process of image analysis.

This adaptive approach to computer vision promises to greatly reduce the number of decisions that image analysts must make per picture, thus improving their ability to deal with a high flow of images. Moreover, the resulting systems should adapt their knowledge to the preferences of individual users in response to feedback from those users. The overall effect should be a new class of systems for image analysis that reduce the workload on human analysts and give them more reliable results, thus speeding the image analysis process.

In the sections that follow, we report initial progress on using machine learning to improve decision making at one stage in an existing image understanding

[*]This research was supported by the Defense Advanced Research Projects Agency under grant N00014-94-1-0746, administered by the Office of Naval Research.

system. We begin by explaining the task domain — identifying buildings in aerial photographs — and then describe the vision system designed for this task. Next we review five well-known algorithms for supervised learning that hold potential for improving the reliability of image analysis in this domain. After this, we report the design of experiments to evaluate these methods and the results of those studies. In closing, we consider related work on learning for image understanding and some directions for future research.

2 Nature of the Image Analysis Task

The image analyst interprets aerial images of ground sites with an eye to unusual activity or other interesting behavior. The images under scrutiny are usually complex, involving many objects arranged in a variety of patterns. A typical image from the Fort Hood RADIUS repository, which contains satellite photographs of a military base, includes buildings in a range of sizes and shapes, major and minor roadways, sidewalks, parking lots, vehicles, and vegetation. A common task faced by the image analyst is to detect change at a site as reflected in differences between two images, as in the number of buildings, roads, and vehicles. This in turn requires the ability to recognize examples from each class of interest. In this paper, we focus on the performance task of identifying buildings in satellite photographs.

Aerial images can vary across a number of dimensions. The most obvious factors concern viewing parameters, such as distance from the site (which affects size and resolution) and viewing angle (which affects perspective and visible surfaces). But other variables also influence the nature of the image, including the time of day (which affects contrast and shadows), the time of year (which affects foliage), and the site itself (which determines the shapes of viewed objects). Taken together, these factors introduce considerable variability into the images that confront the analyst.

In turn, this variability can significantly complicate the task of recognizing object classes. Although a building or vehicle will appear different from alternative perspectives and distances, the effects of such transformations are reasonably well understood. But variations due to time of day, the season, and the site are more serious. Shadows and foliage can hide edges and obscure surfaces, and buildings at distinct sites may have quite different structures and layouts. Such variations serve as mere distractions to the human image analyst, yet they provide serious challenges to existing computer vision systems.

This suggests a natural task for machine learning: given aerial images as training data, acquire knowledge that improves the reliability of such an image analysis system. However, we cannot study this task in the abstract. We must explore the effect of specific induction algorithms on particular vision software. In the next two sections, we briefly review one such system for image analysis, followed by five learning methods that might give it more robust behavior.

3 An Architecture for Image Analysis

Lin and Nevatia [1996] report a computer vision system for the analysis of ground sites in aerial images. Like many programs for image understanding, their system operates in a series of processing stages. Each step involves aggregating lower level features into higher level ones, eventually reaching hypotheses about the locations of buildings. We will consider these stages in the order they occur.

Starting at the pixel level, the system uses an edge detector to group pixels into edgels, and then invokes a line finder to group edgels into lines. Junctions and parallel lines are identified and combined to form three-sided structures or "Us". The algorithm then groups selected Us and junctions to form parallelograms. Each such parallelogram constitutes a hypothesis about the position and orientation of the roof for some building, so we may call this step 'rooftop generation'.

After the system has completed the above aggregation process, a 'rooftop selection' stage evaluates each hypothesis to determine whether that candidate has sufficient evidence to be retained. The aim of this process is to remove hypotheses that do not correspond to actual buildings. Ideally, the system will reject most spurious hypotheses at this point, although a final 'verification' step may still collapse duplicate or overlapping rooftops. This stage may also exclude hypotheses if there exists no evidence of three-dimensional structure, such as shadows and walls.

Analysis of the system's operation suggested that rooftop selection held the most promise for improvement through machine learning, because this stage must deal with many spurious hypotheses. This process takes into account both local and global criteria. Local support comes from features such as lines and corners that are close to a given parallelogram. Since these suggest walls and shadows, they provide evidence that the hypothesis corresponds to an actual building. Global criteria consider containment, overlap, and duplication of hypotheses. Using these evaluation criteria, the set of hypotheses is reduced to a more manageable size.

The individual constraints applied in this process have a solid foundation in both theory and practice. The problem is that we have only heuristic knowledge about how to combine them. Moreover, such rules of thumb are currently crafted by hand, and they do not fare well on images that vary in their global characteristics, such as contrast and amount of shadow. However, methods from machine learning, to which we now turn, may be able to induce better conditions for selecting or rejecting candidate roofs. If these acquired heuristics are more accurate than the existing handcrafted solutions, they will improve the reliability of the rooftop selection process.

4 A Review of Learning Techniques

We can formulate the task of acquiring rooftop selection heuristics in terms of supervised learning. In this process, training cases of some concept are labeled as to their class. In rooftop selection, only two classes exist — rooftop and non-rooftop — which we will refer to as positive and negative examples of the rooftop concept. Each instance consists of a number of attributes and their associated values, along with a class label. These labeled instances constitute training data that are provided as input to an inductive learning routine, which generates concept descriptions designed to distinguish the positive examples from the negative ones. These knowledge structures state the conditions under which the concept, in this case 'rooftop', is satisfied.

For this study, we selected five well-known learning methods: Quinlan's [1993] C4.5, Clark and Niblett's [1989] CN2, Nearest neighbor [e.g., Aha et al., 1992], Naive Bayes [e.g., Langley et al., 1992], and Perceptron learning [e.g., Zurada, 1992]. We chose these methods because they represent a range of representations, performance schemes, and learning mechanisms for supervised concept learning. In addition, they exhibit different inductive biases, meaning that the algorithms acquire certain concepts more easily than others. As a result, they should provide insights about the types of machine learning algorithms that are useful for the rooftop selection task.

C4.5 [Quinlan, 1993] constructs a 'decision tree' from training data. Each nonterminal node in such a tree specifies an attribute, and the emanating links indicate a value (or range of values), whereas terminal nodes specify a class name. Classification occurs by sorting an instance downward through the tree until it reaches a terminal node. Algorithms for inducing decision trees operate by selecting the attribute whose values best discriminate among the classes, partitioning the training data into subsets for each value (or range), then applying the process in turn to each of the resulting subsets. This recursive partitioning process divides the training data into ever smaller sets, until each set contains only one class or no further splits are possible. Some variants, including C4.5, include a 'pruning' stage that cuts back the tree after its construction.

CN2 [Clark and Niblett, 1989, Clark and Boswell, 1991] learns a set of conjunctive rules from training examples. To classify an unknown instance, CN2 determines which rules the instance satisfies. When rules from different classes match an instance, the system uses a probabilistic conflict resolution scheme to select the single best rule. CN2 uses a covering algorithm, much like AQ [Michalski, 1969], that constructs rules one at a time. It specializes a maximally general description until it finds a "best" rule, as determined by some evaluation criterion. CN2 removes those training examples covered by the rule and repeats this process with the remaining examples, creating additional rules until all examples have been covered. The system copes with continuous data by dividing the continuous range into discrete sub-intervals.

Another approach is the nearest neighbor method [e.g., Aha et al., 1991], which uses an 'instance-based' representation of knowledge that simply retains training cases in memory. This approach classifies new instances by finding the 'nearest' stored case, as measured by some distance metric, then predicting the class associated with that case. For numeric attributes, a common metric (which we also use in our studies) is Euclidean distance. In this framework, learning involves nothing more than storing each training instance, along with its associated class. Although this method is quite simple and has known sensitivity to irrelevant attributes, in practice it performs well in many domains. Some versions select the k closest cases and predict the majority class; here we will focus on the 'simple' nearest neighbor scheme, which uses only the nearest case.

A fourth alternative is the naive Bayesian classifier [e.g., Langley et al., 1992], which stores a probabilistic concept description for each class. This description includes an estimate of the class probability and the estimated conditional probabilities of each attribute value given the class. The method classifies new instances by computing the probability of each class using Bayes' rule, combining the stored probabilities by assuming that the attributes are independent given the class and predicting the class with the highest probability. Like nearest neighbor, naive Bayes has known limitations, such as sensitivity to attribute correlations and an inability to represent multiple decision regions, but in practice it behaves well on many natural domains.

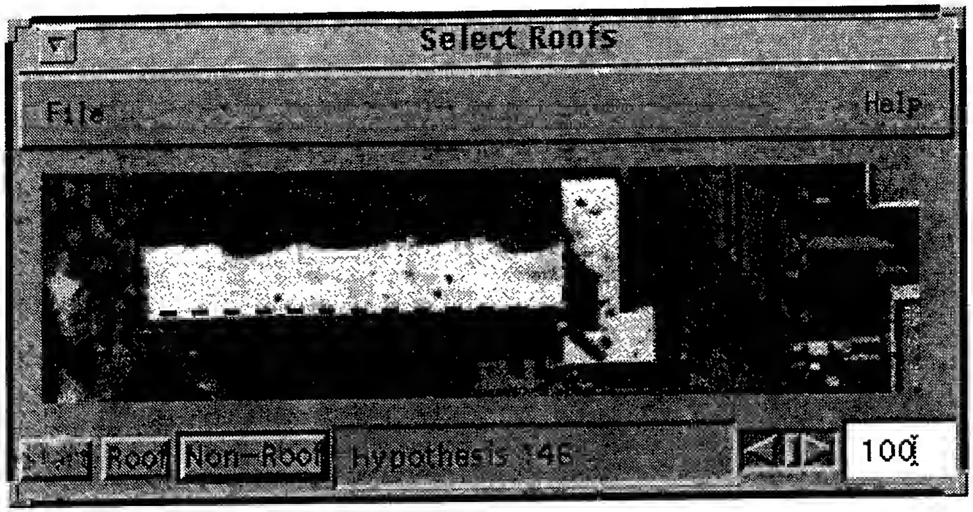


Figure 1: Visualization interface for roof hypotheses.

Finally, the Perceptron learning algorithm [e.g., Zurada, 1992] finds coefficients and a threshold for a linear discriminant function. The algorithm learns these values by applying a hill-climbing technique until it minimizes the error between the desired output and the actual output of the linear discriminant function. To classify unknown instances, the algorithm simply computes the weighted sum using the learned coefficients and the attribute values. If the weighted sum exceeds the threshold, then the system assigns the instance to one class; otherwise, it assigns the instance to the other class. No modifications are needed to handle continuous data (cf. CN2). This algorithm is well-studied and learns the same type of classifier as the Lin/Nevatia [1996] rule, which also takes the form of a linear discriminant function.

5 Representation and Labeling of Rooftop Hypotheses

To apply the above algorithms to the problem of roof hypothesis selection, we selected two data sets derived from aerial images of Fort Hood, Texas. The site contains 29 actual buildings. The first data set, derived from image FHOV1027, consists of 1179 roof hypotheses generated from a nadir view of the site; in contrast, the second data set, derived from image FHOV625, contains 2193 hypotheses produced from an oblique view taken at a different time. Each hypothesis in the two data sets is described in terms of nine features that summarize the evidence gathered from lower levels of analysis; these include evaluation of edge support, corner support, parallel support, orthogonal trihedral vertex support, shadow corner support, gap overlap, displacement of edge support, crossing lines, and existence of junctions. All nine features take on continuous values.

Before we could pass the hypotheses to a learning algorithm, we first had to label each one as either a positive or negative example of the desired concept. To accomplish this task easily, we implemented a visualization system, shown in Figure 1, that displays each roof hypothesis and lets the user label it as positive or negative. There are two problems with this approach. First, the user may have to label thousands of hypotheses, which would be time-consuming and tedious. To reduce the number of hypotheses a human has to label, we implemented a simple pre-screening algorithm that takes user-identified regions of interest (i.e., areas surrounding buildings) and determines how many corners of a rooftop hypothesis fall within this region. For example, if the user specified that two corners must fall within the region, then those hypotheses with fewer than two corners are labeled automatically as negative, and the remaining hypotheses are passed to the visualization system for human classification. For image FHOV1027, the pre-screening step reduced the number of hypotheses that required labeling from 1179 to 257.

The second problem is that the visualization system displays hypotheses in the visual space and not in the attribute space. Early experiences with the visualization system showed it was difficult to judge the quality of hypotheses simply by their visual characteristics (i.e., the set of four lines on the image). Consequently, we needed a feedback mechanism to show how the hypothesis looked in the visual and attribute spaces.

To address this problem, we incorporated a simple learning algorithm into the visualization system that uses a nearest neighbor classifier and its past experience to classify a new hypothesis. The system

Table 1: Experimental results for within-image learning using data from two Fort Hood images.

	(a) Image FHOV1027			(b) Image FHOV625		
	Positive Accuracy	Negative Accuracy	Overall Accuracy	Positive Accuracy	Negative Accuracy	Overall Accuracy
Lin/Nevatia	48.80±1.4	100.00±0.0	82.91±0.6	*69.64±1.8*	100.00±0.0	*91.58±0.3*
Naive Bayes	*74.56±1.5*	83.84±0.7	82.28±0.5	59.72±1.6	91.40±0.7	87.82±0.6
Nearest neighbor	60.08±1.9	93.00±0.6	87.50±0.5	54.32±1.9	94.00±0.3	89.63±0.4
C4.5 (w/pruning)	58.12±3.7	94.00±0.5	87.95±0.5	39.24±1.8	96.56±0.6	90.28±0.4
CN2	47.84±1.4	98.44±0.2	*89.99±0.4*	0.00±0.0	100.00±0.0	89.20±0.3
Perceptron	0.00±0.0	100.00±0.0	83.20±0.5	1.60±0.9	100.00±0.0	89.37±0.3

displays hypotheses classified to be roofs as green rectangles, non-roofs as blue rectangles, and atypical hypotheses as red rectangles. The user can set a "sensitivity threshold" that affects how distant from previous instances a hypothesis must be before it is labeled atypical. As the visualization system gains experience, it displays fewer and fewer atypical hypotheses. After the system classifies and displays a hypothesis, the user either confirms or overrides the system's decision. By the end of the labeling session, the user typically confirms most of the decisions made by the system.

6 Within-Image Learning

A typical machine learning experiment manipulates one or more independent variables and evaluates the effect of this manipulation by measuring one or more dependent variables. Since our hypothesis was that machine learning would produce classifiers that perform as well or better than handcrafted knowledge, the natural independent variable was the classifier used to label hypotheses, in particular whether one used the Lin/Nevatia heuristic versus a learned classifier. Because we were also interested in the behavior of different methods, we compared the Lin/Nevatia scheme to all the learning methods described in Section 4. The obvious dependent variable is overall accuracy of the learned knowledge on unseen instances, computed as the percentage of correctly classified test instances. However, the cost of discarding a hypothesis that corresponds to an actual rooftop is more expensive than retaining a hypothesis that does not. Consequently, we also measured accuracy on both the positive and negative instances.

Our first study took the form of a common type of experiment found in the machine learning literature and involved forming multiple partitions of a data set into training set/testing set pairs. For a given pair, one applies the learning algorithm to the training set and evaluates the acquired knowledge on the test set. Averaging results across all pairs provides a good estimate of the accuracy for a given algorithm, since it minimizes the effect of unrepresentative samples in the training or testing sets. We conducted this experiment using data derived from image FHOV1027 and image FHOV625. We randomly generated 25 partitions of the data set. For each partition, the training set consisted of 60% of the original data set, while the testing set consisted of the remaining 40%; we then applied all of the learning algorithms to each train/test pair. For this experiment, we used the MLC++ library of machine learning programs [Kohavi et al., 1996].

For comparison, we also ran the Lin/Nevatia [1996] selection criterion on each test set and computed its average accuracy. Because its authors constructed this heuristic manually, no training was involved. Of course, to the extent that Lin and Nevatia modified it in response to its behavior, they did 'train' their heuristic, but this training was conducted using separate images from a 'model board', which were photographs of a scale-model site for which ground truth was known. This does not provide the best comparison possible. However, our concern here is not with the origin of their heuristic but with trying to improve upon it.

Table 1 shows the results of the within-image learning experiments using data generated from images FHOV1027 and FHOV625. The table presents results for each image in terms of average positive, negative, and overall accuracy, with 95% confidence intervals. The highest accuracies for the positive class and overall are indicated by italics. For image FHOV1027 (Table 1a), naive Bayes performed the worst overall, but had the best performance for the positive class. On the other hand, CN2 performed best overall, but performed poorly in terms of positive accuracy. For image FHOV625 (Table 1b), the Lin/Nevatia classifier performed best in terms of overall accuracy and positive accuracy, whereas naive Bayes was second, but only in terms of positive accuracy.

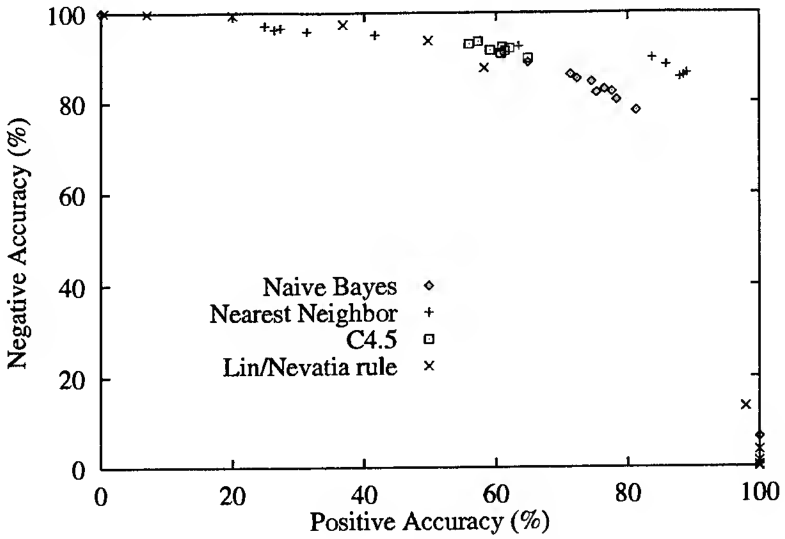


Figure 2: ROC curve for image FHOV1027.

To provide a context for these results, consider the performance of a frequency-based classifier that always predicts the most frequent class. Given the data generated from images FHOV1027 FHOV627, this rule would always predict a negative instance, resulting in a positive accuracy of 0%, a negative accuracy of 100%, and an overall accuracy reflecting the percentage of negative instances in the data set, or 87.1%. An extreme bias toward positive accuracy would lead one to always predict a positive instance, resulting in a positive accuracy of 100%, a negative accuracy of 0%, and an overall accuracy reflecting the percentage of positive instances in the data set, or 12.9%. The Lin/Nevatia rule, naive Bayes, and nearest neighbor all find comfortable trade-offs between positive and negative accuracy. These results are roughly consistent with our hypothesis, but they are hardly conclusive.

An important thing to note about the results on these two images is how the relative position of the machine learning methods stayed essentially the same. The one exception, on the second image, was that CN2 performed slightly worse than the Perceptron learning algorithm. This consistency is encouraging because it suggests that we can identify a single algorithm that will perform well across a range of images. Hopefully, experiments with more images will demonstrate the same trend.

7 The Cost of Misclassification

Given the above results, naive Bayes and nearest neighbor show promise of being comparable to the Lin/Nevatia rule in terms of accuracy on the positive instances. Ideally, we would like to improve upon their positive accuracy without losing much negative accuracy. To achieve this affect, the learning algorithm must be biased in favor of positive accuracy, but most machine learning methods do not provide ways to accomplish this. Pazzani et al. [1994] have done some preliminary work along these lines, which they describe as addressing differing costs of error types. The basic idea is to change the way the algorithm treats positive instances relative to negatives, either during the learning process or at the time of testing.

This approach should also give us more principled comparisons among the various learning methods. By systematically varying the relative costs, we can generate a Receiver Operating Characteristic (ROC) curve, which graphs negative accuracies as a function of positive accuracies. The ROC curve for each algorithm provides a cost-independent summary of its behavior, with curves that cover larger areas being generally better. This suggests a revision of our hypothesis from the previous section: as one varies error costs, machine learning will produce ROC curves with equal or larger areas than that covered by the Lin/Nevatia classifier.

To test this hypothesis, we implemented or obtained cost sensitive versions of the best performing algorithms from the previous experiments, namely naive Bayes, nearest neighbor, and C4.5. We defined a cost on the range [−1.0, 1.0], where a negative setting means that mistakes on the positive class are more costly and a positive setting means that mistakes on the negative class are more costly. Although this formulation differs slightly from Pazzani et al.'s [1994], it is equivalent for two-class problems.

For naive Bayes, we used the cost sensitive measure to adjust the Bayesian posterior probabilities [Duda

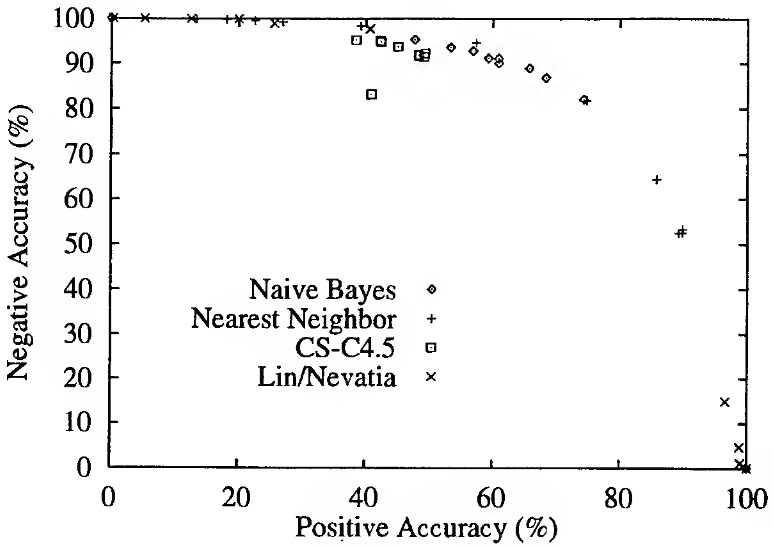


Figure 3: ROC curve for image FHOV625.

and Hart, 1973, Pazzani et al., 1994]. Specifically, the modified algorithm alters the posterior probability of the preferred class so that it becomes more probable. The class whose posterior probability is modified, and the degree to which it is changed, depends on the sign and magnitude of the cost metric.

We introduced a cost sensitive measure into the nearest neighbor algorithm by adjusting the distance from an unknown instance to its closest neighbor for each class, positive and negative. After making this adjustment based on the sign and magnitude of the cost measure, the classification process proceeds normally, assigning the class label of the "closest" neighbor. This modification also works for versions of the algorithm that consider more than one neighbor when classifying unknown instances.

We were able to make similar modifications to the Lin/Nevatia classifier [1996] for purposes of comparison. Since this classifier is a linear discriminant function, we adjusted the threshold such that the decision boundary is closer to the hypothetical cluster of positive examples or the cluster of negative examples. The direction and degree to which we adjust the threshold is again dependent on the sign and magnitude of the cost measure.

We also obtained a cost sensitive version of C4.5 [Grimmer, 1997], which takes a different approach to learning minimum cost classifiers than ours. Rather than using cost sensitive measures in the testing phase, it takes cost into account during the learning phase when it prunes the decision tree [Breiman et al., 1984]. Briefly, the pruning algorithm selects the least costly of three actions for each subtree: do nothing (i.e., leave the subtree unpruned); replace the subtree with a leaf node and assign the majority class label of the subtree to the leaf node; and replace the subtree with the subtree of one of its children. Costs for classes in this version of C4.5 are expressed on the integer range [0, ∞].

To investigate the effect of misclassification costs, we conducted an experiment using the cost sensitive versions of nearest neighbor, naive Bayes, C4.5, and the Lin/Nevatia classifier. One condition used data from image FHOV1027, while another used data from image FHOV625. For each algorithm and image, we varied the cost metric and measured the resulting positive and negative accuracy for ten runs; each run involved partitioning the data set randomly into training (60%) and testing (40%) sets. For each cost setting and each classifier, we plotted the average true negative rate (i.e., accuracy on negative cases) against the true positive rate (i.e., accuracy on positive cases). Figures 2 and 3 show the ROC curves that resulted from this procedure for images FHOV1027 and FHOV625, respectively.

The most notable aspect of Figure 2 is that the curves for most of the learned classifiers are nearly identical with that for the Lin/Nevatia rule, giving support for our hypothesis that machine learning can at least match handcrafted knowledge. However, portions of the ROC curve for cost sensitive nearest neighbor stand out as substantially higher than others, suggesting that this method outperforms both the Lin/Nevatia and the alternative learning schemes. This difference was not apparent from our earlier experiment, and shows the clear advantage of using ROC curves to compare methods in cost sensitive domains.

Inspection of Figure 3 reveals additional support for our basic hypothesis, since again the Lin/Nevatia

Table 2: Experimental results for between-image learning using data from two Fort Hood images.

	(a) Train FHOV1027/Test FHOV625			(b) Train FHOV625/Test FHOV1027		
	Positive Accuracy	Negative Accuracy	Overall Accuracy	Positive Accuracy	Negative Accuracy	Overall Accuracy
Lin/Nevatia	*68.00±0.0*	100.00±0.0	*91.52±0.0*	50.00±0.0	100.00±0.0	83.12±0.0
Naive Bayes	33.72±0.5	98.12±0.1	91.32±0.1	*86.44±0.7*	64.12±1.4	67.88±1.0
Nearest neighbor	25.96±0.8	97.16±0.2	89.48±0.2	71.36±1.3	80.60±0.5	79.12±0.3
C4.5 (w/pruning)	24.56±1.5	98.00±0.3	89.98±0.2	58.60±2.5	84.40±0.9	80.10±0.7
CN2	60.00±0.0	100.00±0.0	89.60±0.0	0.00±0.0	100.00±0.0	83.30±0.0
Perceptron	0.00±0.0	100.00±0.0	89.15±0.0	1.28±0.8	100.00±0.0	*83.47±0.1*

curve runs the same course as those for most learned classifiers. But for this image, nearest neighbor does no better than the other induction algorithms, and portions of the C4.5 curve appear substantially worse. Clearly, we should replicate these results on more images, but the results so far are generally encouraging.

8 Between-Image Learning

We geared our final study more toward the goals of image analysis. Recall that our motivating problem is the large number of images that must be processed. In order to alleviate the burden on the image analyst, we want to apply knowledge learned from some images to many other images. We have already noted that several dimensions of variation pose problems to transferring learned knowledge to new images. For example, one viewpoint of a given site can differ from other viewpoints of the same site in both orientation and angle from the perpendicular. We need to better understand how the knowledge learned from one image generalizes to other images that differ along these dimensions. Images taken at different times and images of different sites present similar issues. Our hypothesis here was a stronger version of earlier ones: classifiers learned from one image would perform as well or better on unseen images than handcrafted classifiers. However, we also expected that such between-image learning would give less impressive results than the within-image situation.

To test our predictions, we simply examined generalization across the two images, which differ both in viewing angle and in time. Clearly, future experiments should systematically vary each of these independent variables, to determine their individual effect on transfer, but the current comparison should give us some insight into how well learned knowledge carries across images. For each learning algorithm, we conducted 25 runs in which 60% of the hypotheses from image FHOV1027 served as the training set and all hypotheses from image FHOV625 as the test set. In addition, we carried out the same procedure using 60% of the data from image FHOV625 for training and all of the data from image FHOV1027 for testing. For this experiment, we again took advantage of the MLC++ library of machine learning programs [Kohavi et al., 1996].

The results for this between-image experiment appear in Table 2. For the first condition (Table 2a), the Lin/Nevatia rule performed the same as in the second within-image learning condition (Table 1b), since the same data was used for testing. Recall that no comparable notion of generalization applies to the Lin/Nevatia classifier since it was handcrafted for unrelated data. The overall performance of naive Bayes was very high, but its predictive accuracy on the positive class was much less than for CN2 and the Lin/Nevatia classifier. CN2 achieved a respectable accuracy on the positive examples for this problem, in contrast to its performance in the experiment on within-image learning.

For the second between-image condition (Table 2b), the Lin/Nevatia rule again performed as it did in the first within-image condition (Table 1a), since the same data was used for testing. Although the Perceptron algorithm achieved the highest overall accuracy, CN2 and the Lin/Nevatia rule performed nearly as well. In terms of positive accuracy, naive Bayes and nearest neighbor performed highest; yet in terms of overall accuracy, these two methods fared the worst.

However, evaluating our hypothesis about between-image generalization involves comparing with accuracies from the within-image study, and here the results are mixed. Table 1b reports performance when we trained and tested the learning methods on image FHOV625, whereas Table 2a reports performance for training on FHOV1027 and testing on FHOV625. As predicted, the positive accuracies for naive Bayes, nearest neighbor, and C4.5 in the between-image case are lower than for the within-image condition, suggesting considerably less generalization across images than within them.

But, comparing Table 1a (training and testing on FHOV1027) with Table 2b reveals a different story. Here the positive accuracies are substantially higher for the between-image condition, suggesting that generalization was actually better across images than within them. This finding is more encouraging but runs counter to our expectation that training and testing on different images would be a more difficult learning task than training and testing on the same image. The results are further confused by CN2's behavior, which showed higher accuracies than expected in Table 2a but not in Table 2b.

Clearly, the results from this experiment are inconclusive, and we suspect that both the large decrease in FHOV1027 accuracies and the increase in FHOV625 accuracies are artifacts due to the particular distribution of positive and negative instances in these data sets. Repeating the study using cost sensitive versions of the learning algorithms, and calculating ROC curves for each pair of training and test images, should reveal the generalization for each case in a distribution-free manner. In this framework, we would expect that the area under the ROC curve for between-image learning on a given test image and a given method will be less than the area for within-image learning on the same image and method.

In summary, our experiments have revealed some factors that influence performance on the rooftop selection task — the learning algorithm used to acquire knowledge, the relative cost of classification errors, and the nature of the images themselves. We showed that, at least for positive accuracies, the naive Bayesian and nearest neighbor classifiers closely approach (and sometimes exceed) the handcrafted Lin/Nevatia heuristic. Although it is difficult to identify the conditions under generalization occurs between image, we have available a methodology that will help us investigate this issue further.

9 Related Work on Visual Learning

Research on learning in computer vision has become increasingly common in recent years. Papers by Conklin [1993], Sengupta and Boyer [1993], Cook et al. [1993], and Provan et al. [1996] all describe approaches to learning three-dimensional descriptions for use in object recognition. Another approach [e.g., Gros, 1993, Pope and Lowe, 1996] instead learns characteristic views for use in recognition, while still others focus on learning the appearances of objects or scenes [e.g., Nayar et al., 1996, Pomerleau, 1996, Viola, 1993].

Most work on visual learning ignores the importance of misclassification costs, but our work along these lines has some precedents. In particular, Draper et al. [1994] incorporate the cost of errors into their algorithm for constructing and pruning multivariate decision trees. They tested this approach on the task of labeling pixels from outdoor images for use by a road-following vehicle. They determined that, in this context, labeling a road pixel as non-road was more costly than the reverse, and showed experimentally that their method could reduce such errors on novel test pixels.

However, like much of the research on visual learning, Draper et al.'s work focused on image processing in complex scenes at eye level. One exception is Connell and Brady's [1987] work on learning structural descriptions of airplanes from aerial views. Their method converted training images into semantic networks, which it then generalized on comparison with descriptions of other instances. However, Connell and Brady do not appear to have tested experimentally their algorithm's ability to accurately classify objects in new images.

Draper [1996] reports a more careful study of learning in the context of analyzing aerial images. His approach adapts methods for reinforcement learning to assign credit in multi-stage image processing (for software similar to the Lin/Nevatia system), then uses an induction method (backpropagation in neural networks) to learn conditions on operator selection. He presents initial results on a RADIUS task that also involves the detection of roofs.

Our framework shares some features with Draper's approach, but assumes that learning is directed by feedback from a human expert. We predict that our supervised method will be more computationally tractable than his use of reinforcement learning, which is well known for its high complexity. Our approach does require more interaction with users, but we believe this interaction will be unobtrusive if cast within the context of an advisory system for image analysis.

10 Concluding Remarks

Although our initial studies have provided some insights into the role of machine learning in image analysis, much still remains to be done. For example, we may want to consider alternate measures of classification accuracy that take into account the presence of multiple valid hypotheses for a given rooftop. Classifying one of these hypotheses correctly is sufficient. In addition, although the rooftop selection stage was a natural place to start in applying our methods, we are also interested in working at both earlier and later levels of the process. Note that the goal here is not only to increase classification accuracy, which could be handled entirely by hypothesis selection, but to reduce the complexity of processing by removing bad hypotheses before they are aggregated into larger structures. With this aim in mind, we plan to extend our work to apply at all levels of the image understanding process.

We must address a number of issues before we can apply machine learning to other stages. One involves identifying the cost of different errors at each level, and taking this into account in our modified induction algorithms. Another concerns whether we should use the same induction algorithm at each level or use different methods at each stage. We should also explore using a number of learning methods in combination, either averaging their predictions (as in work on 'ensembles') or cascading the results (as in work on 'boosting').

As we mentioned earlier, in order to automate the collection of training data for learning, we also hope to integrate learning routines into the Lin/Nevatia software. This system was not designed initially to be interactive, but we would like to modify it so that the image analyst can accept or reject recommendations made by the image understanding system, generating training data in the process. At intervals the system would invoke its learning algorithms, producing revised knowledge that would alter the system's behavior in the future and, hopefully, reduce the user's need to make corrections. The interactive labeling system described in Section 5 could serve as an initial model for this interface.

In conclusion, our studies suggest that machine learning has an important role to play in improving the accuracy, and thus the robustness, of image analysis systems. .However, we need additional experiments to give better understanding of the factors affecting between-image transfer and we need to extend learning to additional levels of the image understanding process. Also, before we can develop a system that truly aids the human image analyst, we must develop and implement unobtrusive ways to collect training data to support learning.

Acknowledgements

The authors thank Ram Nevatia, Andres Huertas and Andy Lin for their assistance in obtaining the images and data used for experimentation and providing valuable comments and advice. We would also like to thank Wayne Iba for his help with naive Bayes and Udo Grimmer for providing the cost sensitive version of C4.5. This research was conducted at the Institute for the Study of Learning and Expertise and in the Computational Learning Laboratory, Center for the Study of Language and Information, at Stanford University.

References

[Aha et al., 1991] Aha, D.W., Kibler, D. and Albert, M.K. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Breiman et al., 1984] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. *Classification and regression trees.* Belmont, CA: Wadsworth, 1984.

[Clark and Boswell, 1991] Clark, P. and Boswell, R. Rule induction with CN2: some recent improvements. *Machine Learning — Proceedings of the Fifth European Conference (EWSL-91)*, 151–163. Berlin: Springer-Verlag, 1991.

[Clark and Niblett, 1989] Clark, P. and Niblett, T. The CN2 induction algorithm. *Machine Learning*, 3:261–284, 1989.

[Conklin, 1993] Conklin, D. Transformation-invariant indexing and machine discovery for computer vision. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 10–14. Menlo Park, CA: AAAI Press, 1993.

[Connell and Brady, 1987] Connell, J.H. and Brady, M. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[Cook et al., 1993] Cook, D., Hall, L., Stark, L. and Bowyer, K. Learning combination of evidence functions in object recognition. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 139–143. Menlo Park, CA: AAAI Press, 1993.

[Draper et al., 1994] Draper, B.A., Brodley, C.E. and Utgoff, P.E. Goal-directed classification using linear machine decision trees. *IEEE Transactions on PAMI*, 16.9:888–893, 1994.

[Draper 1996] Draper, B. Learning grouping strategies for 2D and 3D object recognition. *Proceedings of the Image Understanding Workshop*, 1447–1454. San Francisco, CA: Morgan Kaufmann, 1996.

[Duda and Hart, 1973] Duda, R.O., and Hart, P.E. *Pattern classification and scene analysis.* New York, NY: Wiley, 1973.

[Grimmer, 1997] Grimmer, U. Personal communication. 1997.

[Gros, 1993] Gros, P. Matching and clustering: Two steps towards automatic object model generation in computer vision. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 40–44. Menlo Park, CA: AAAI Press, 1993.

[Kohavi et al., 1996] Kohavi, R., Sommerfield, D. and Dougherty, J. Data mining using MLC++: a machine learning library in C++. *Proceedings of IEEE Tools with Artificial Intelligence*, 234–245, 1996 (http://www.sgi.com/Technology/mlc).

[Langley et al., 1992] Langley, P., Iba, W. and Thompson, K. An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 223–228. Menlo Park, CA: AAAI Press, 1992.

[Langley and Simon, 1995] Langley, P. and Simon, H.A. Applications of machine learning and rule induction. *Communications of the ACM*, 38 (November) 55–64, 1995.

[Lin and Nevatia, 1996] Lin, C. and Nevatia, R. Building detection and description from monocular aerial images. *Proceedings of the Image Understanding Workshop*, 461–468. San Francisco, CA: Morgan Kaufmann, 1996.

[Michalski, 1969] Michalski, R.S. On the quasiminimal solution of the general covering problem. *Proceedings of the 5th International Symposium on Information Processing*, Vol. A3, 125–128, 1969.

[Nayar et al., 1996] Nayar, S.K., Murase, H. and Nene, S.A. Parametric appearance representation. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 131–160. New York: Oxford University Press, 1996.

[Pazzani et al., 1994] Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. and Brunk, C. Reducing misclassification costs. *Proceedings of the Eleventh International Conference on Machine Learning*, 217–225. San Francisco, CA: Morgan Kaufmann, 1994.

[Pomerleau, 1996] Pomerleau, D. Neural network vision for robot driving. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 161–181. New York: Oxford University Press, 1996.

[Pope and Lowe, 1996] Pope, A.R. and Lowe, D.G. Learning probabilistic appearance models for object recognition. In Nayar, S.K. and Poggio, T., eds., *Early visual learning*, 67–97. New York: Oxford University Press, 1996.

[Provan et al., 1996] Provan, G., Langley, P. and Binford, T.O. Probabilistic learning of three-dimensional object models. *Proceedings of the Image Understanding Workshop*, 1403–1413. San Francisco, CA: Morgan Kaufmann, 1996.

[Quinlan, 1993] Quinlan, J.R. *C4.5: Programs for machine learning.* San Francisco, CA: Morgan Kaufmann, 1993.

[Sengupta and Boyer, 1993] Sengupta, K. and Boyer, K.L. Incremental model base updating: Learning new model sites. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 1–5. Menlo Park, CA: AAAI Press, 1993.

[Viola, 1993] Viola, P.A. Feature-based recognition of objects. *Working Notes of the AAAI Fall Symposium on Machine Learning in Computer Vision*, 60–64. Menlo Park, CA: AAAI Press, 1993.

[Zurada, 1992] Zurada, J.M. *Introduction to artificial neural systems.* St. Paul, MN: West Publishing, 1992.

AUTOMATIC POPLUATION OF GEOSPATIAL DATABASES (APGD) TECHNICAL PAPERS

Geospatial Registration

Rakesh Kumar* Harpreet S. Sawhney ; Jane C. Asmuth
David Sarnoff Research Center, CN5300, Princeton, NJ 08530 {sawhney,kumar}@sarnoff.com

Abstract

The ability to locate scenes and objects visible in a video/image frame to their corresponding locations and coordinates in a reference coordinate system will be important in visually-guided navigation, surveillance and monitoring systems of the future. Aerial video is rapidly emerging as a low cost, widely used source of imagery for mapping, surveillance and monitoring applications. A database of reference imagery and the associated geo-coordinates (e.g. latitude/longitude) is often available for locales that are surveyed using current videos. However, a key technical problem of locating objects and scenes in a video to their geo-coordinates needs to be solved in order to ascertain the geo-location of objects seen from the camera platform's current location. In this paper we present the design of a system and key algorithms for the problem of accurate mapping between camera coordinates and geo-coordinates, called *geo-spatial registration.* Current systems for geo-location use the position and attitude information for the moving platform in some fixed world coordinates to locate the video frames in the reference database. However, the accuracy achieved is only of the order of 10's to 100's of pixels. Our approach utilizes the imagery and terrain information contained in the geo-spatial database to precisely align dynamic videos with the reference imagery and thus achieves a much higher accuracy. Applications of geo-spatial registration include text/graphical/audio annotations of objects of interest in the current video using the stored annotations in the reference database. These applications extend beyond aerial videos into the challenging domain of video/image-based map and database indexing of arbitrary locales, like cities and urban areas.

1 Introduction

Aerial video is rapidly emerging as a low cost, widely used source of imagery for mapping, surveillance and monitoring applications. Visible and infrared (IR) video cameras are increasingly deployed on airborne platforms, both manned and unmanned, to provide observers with a real time view of activity and terrain. However, there remains a key technical problem in the use of video from moving vehicles: determining how the locations of objects in a video display relate to the geographic locations of these objects on the observed scene. This relationship between image coordinates and geographic coordinates must be known before actions can be undertaken based on the video. This mapping between camera coordinates and ground coordinates, called geospatial registration, depends both on the location and orientation of the camera and on the distance and topology of the ground. Camera location continually changes as the airborne camera moves. Rough geospatial registration can be derived from camera telemetry data (GPS location of the aircraft and orientation of the camera) and digital terrain map data (from a database) [3]. This form of registration is the best available in fielded systems today but does not provide the precision needed for many tasks. Higher precision will be achieved by correlating (and registering) observed video frames in real time to stored references imagery. The reference imagery includes previously collected satellite images that have been precisely aligned to map coordinates. An application of geospatial registration may be to annotate objects observed in the video by their names or overlay maps, boundaries and other graphical features over the video imagery.

In this paper, we present the design of a system for accurate geospatial registration. We then present the details and results of some of the key algorithms [7] we have developed in our laboratory towards implementing the overall system. This work is in contrast with the previous body of work based on *still* imagery exploitation using site models [9].

1.1 Components of a Geo-registration System

A schematic of the geo-registration and annotation concept is shown in Figure 1. The figure shows a mobile platform capturing current videos of a locale. After geo-registration to the reference imagery database, the footprints of the video are shown overlaid on the reference imagery, and latitude/longitude/height of points of interest are retrieved based on geo-registration and are overlaid on the relevant points on the video frame.

Figure 2 shows the functional layout of the major blocks and flow of information in the geospatial registration indexing, alignment, annotation display, and synthetic view creation system. The system can broadly be divided into the following components:

- Geo-referencing/location module for locating video frames within an imagery and multi-

*The work reported here was funded in part by the National Information Display Laboratory, Princeton, New Jersey

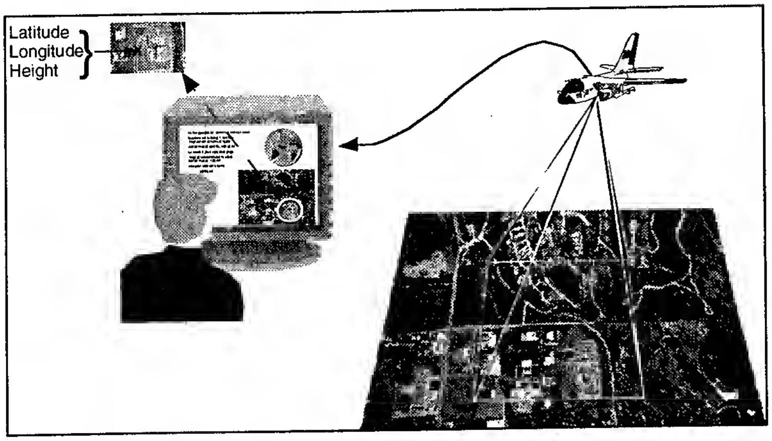


Figure 1: A schematic of the Geo-registration concept.

modal annotated database that is registered with geographical coordinates.

- Annotation overlay module that accesses the relevant annotations using the geo-location module and overlays these on video frames.
- Database of reference imagery, DEM (Digital Elevation Map) and multi-modal annotations such as graphics, text, audio, maps etc.
- New view generation module that can show views of the reference database from the vantage point of the video capture platform.

The engineering support data (ESD: GPS, camera look angle etc.) supplied with the video is decoded to define the camera model (position and attitude) with respect to the reference database. The camera model and scene data base is used to apply an image perspective transformation to create a set of synthetic reference images from the perspective of the sensor. This set of reference images is fed to the real-time video processing system to be used for indexing and fine alignment[1]. Simultaneously, the system indexes into the geospatial feature data base according to the geo-coordinate footprint of sensor and extracts candidate annotations for overlay. These annotations are then fed to the real-time video processing system. The real-time system computes the precise locations for overlaying the annotations on live video. The annotations are overlaid by the video mixer on the analysts reference overview image monitor window with attached sound/text references for point and click recall. Finally, the current estimate of sensor attitude and position is updated based upon results of matching from real-time registration module. This information is used to generate new reference images to support matching based upon new estimates of sensor position and attitude and the whole process is iterated.

1.2 From Videos to Mosaics to Geo-registration

Given that ESD on its own will not be reliable in associating objects seen in videos to their corresponding geo locations, we utilize the precision in localization afforded by the alignment of rich visual attributes typically available in video imagery. For aerial surveillance scenarios, often a reference image database in geo-coordinates along with the associated DEM maps and annotations is available. The visual features available in the reference imagery database are correlated with those in video imagery to achieve an order of magnitude improvement in alignment in comparison to purely ESD based alignment. Our approach uses the ESD to generate initial hypothesis for the geo-located locale of interest for the current video data. The initial hypothesis is typically a section of the reference imagery warped and transformed so that it approximates well the visual appearance of the relevant locale from the viewpoint specified by the ESD. Subsequently, precise sub-pixel alignment between the video frame and reference imagery corresponding to the relevant locale is used for precise geo-location of the video frame. The position and attitude information provided by ESD may be good enough to generate a 3Kx3K section of the relevant reference imagery with a resolution of about 1-2 feet per pixel. This clearly will not be adequate for overlaying potentially complex annotations like maps and object models. Accurate sustained overlays over time on the video imagery require constant maintenance of precise alignment between the geo-referenced annotations and video frames. The process of precise alignment of video to reference imagery itself needs to be divided into three main steps:

[1] The alignment step does not require accurate knowledge of the video camera's calibration parameters.

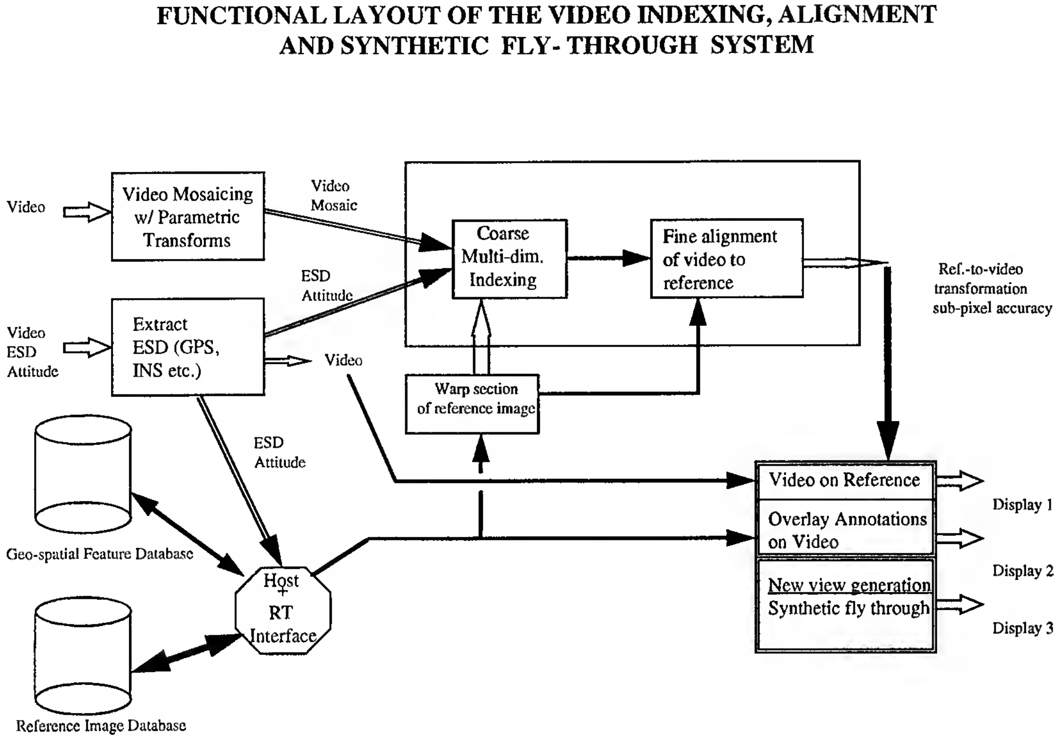

FUNCTIONAL LAYOUT OF THE VIDEO INDEXING, ALIGNMENT AND SYNTHETIC FLY- THROUGH SYSTEM
Video
Video Mosaicing w/ Parametric Transforms
Video Mosaic
Coarse Multi-dim. Indexing
Fine alignment of video to reference
Ref.-to-video transformation sub-pixel accuracy
ESD Attitude
Video ESD Attitude
Extract ESD (GPS, INS etc.)
Video
Warp section of reference image
ESD Attitude
Video on Reference
Overlay Annotations on Video
New view generation Synthetic fly through
Display 1
Display 2
Display 3
Geo-spatial Feature Database
Host RT Interface
Reference Image Database

Figure 2: Functional Specification Block Diagram.

- Video-to-video frame alignment and mosaic creation.
- Coarse indexing/pruning of match locations in the reference imagery to eliminate all but the best possible location of the video imagery in the reference to with a few pixels of its true location,
- Precise parametric alignment with and/or without the DEM map to obtain sub-pixel localization of the video frame in reference imagery.
- Video-to-reference tracking based alignment mode.

2 Lens Distortion Corrected Video Mosaicing

Video frames are typically acquired at 30 frames a second and contain a lot of frame-to-frame overlap. For typical altitudes and speeds of airborne platforms, the overlaps may range from 4/5 to 49/50th of a single frame. Therefore, conversion of video frames to video mosaics is an efficient way to keep up with the incoming video stream. We exploit the redundancy in video frames by aligning successive video frames with low order parametric transformations like translation, affine and projective transformations. The frame-to-frame alignment parameters enable the creation of a single extended view mosaic image that authentically represents all the information contained in the aligned input frames in a compact image. For instance, typically 30 frames of standard NTSC resolution (720x480) containing about 10M pixels may be reduced to a single mosaic image containing only about 200K to 2M pixels depending on the overlap between successive frames. The video mosaic is subsequently available for georeferencing and location.

In the work reported here, instead of using affine transformations as in [5, 10], the frame-to-frame alignment process has been changed in two significant ways: (i) we compute the projective transformation [14], and (ii) the mosaicing process is extended to handle unknown lens distortion present in the image.

The affine transform models the image projection used creating the aerial video as an orthographic projection, while the projective transform models the image projection as an perspective projection. Since, our final goal is to precisely align the video

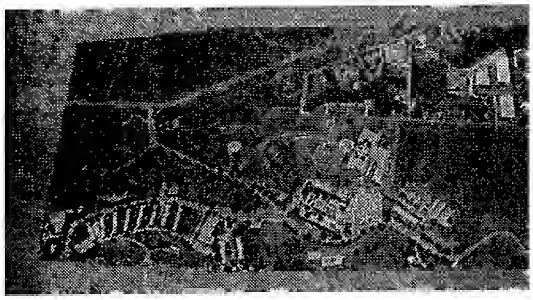


Figure 3: Projective transformation based video mosaic for the oblique video.

mosaics with the reference imagery, the video mosaic is created by using the projective transformation. An example of such a mosaic is shown in Figure 3. Note, in the video mosaicing process we are not taking into account 3D parallax. However, in the overall frame work, a new mosaic is constructed every second or so. In typical scenarios, the 3D parallax observed in the aerial video for such small time frames is very minimal.

Often in aerial video streams, the lens distortion parameters must be explicitly modeled in the estimation process. A fundamental assumption made in the earlier work on mosaicing was that one image could be chosen as the reference image and the mosaic would be constructed by merging all other images to this reference image. However, in the case when lens distortion is present, this is not true. We extend the direct estimation algorithms to use a reference coordinate system but not a reference image. We compute the motion parameters which warp all images to a virtual image mosaic in this reference coordinate system. Each pixel in this virtual image mosaic may be is predicted by intensities from more than one image. The error measure we minimize is the sum of the variances or predicted pixel intensities at each pixel location summed over the virtual image. In order to compute the correspondences and the unknown parameters simultaneously, we formulate an error function that minimizes the variance in intensities of a set of corresponding points in the images, that map to the same ideal reference coordinate. Formally, the unknown projective transformation parameters for each frame, $\mathbf{A}^1 \ldots \mathbf{A}^N$, and the lens distortion parameter, γ_1 are solved for through:

$$\min_{\mathbf{A}^1 \ldots \mathbf{A}^N, \gamma_1} \sum_{\mathbf{p}} \frac{1}{M(\mathbf{p})} \sum_i (I_i(\mathbf{p}^\mathbf{i}) - \bar{I}(\mathbf{p}))^2, \quad (1)$$

where point $\mathbf{p}^\mathbf{i}$ in frame i is a transformation of a point $\mathbf{p}$ in the reference coordinate system, $\bar{I}(\mathbf{p})$ is the mean intensity value of all the $\mathbf{p}^\mathbf{i}$'s that map to $\mathbf{p}$, and $M(\mathbf{p})$ is a count of all such $\mathbf{p}^\mathbf{i}$'s. Therefore, given a point $\mathbf{p}$ in the reference coordinates, each term in the sum in Equation 1 is the variance of all the intensity values at points $\mathbf{p}^\mathbf{i}$ that map to point $\mathbf{p}$. An example of alignment and mosaic construction using this process is shown in the Figures 4 and 5.

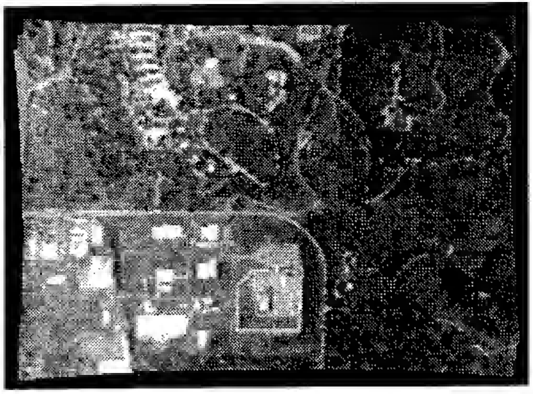


Figure 5: Lens distortion corrected video mosaic of three frames of an aerial video.

3 Coarse Indexing/Matching

The video mosaics created at regular intervals (typically 1 each second) need to be geo- referenced with the reference database. Given the size of the mosaics and the relevant piece of the reference imagery, for real time constraints, the process of locating the video mosaic within the reference coordinates needs to be divided into a coarse indexing/matching step and a precise alignment step. The coarse indexing step locates a video mosaic within a reference image using visual appearance features. In principle, one could exhaustively correlate the intensities in the video mosaic and the reference image at each pixel and find the best match. However, due to the uncertainties in viewpoint from ESD and due to real changes in appearance between the reference imagery and the current video, it may not be possible to directly correlate intensities in the two images. The real changes in appearance may be due to change of reflectance of objects and surfaces in the scene (e.g. summer to fall) and due to difference in illumination between the reference and the video imagery. Changes in appearance due to viewpoint are accounted for to a large extent by the process of warping the reference image to the ESD viewpoint. However, for robust matching and localization, indexing and matching needs to be resilient to uncertainties in ESD and to real changes.

We propose to solve this problem by computing features at multiple scales and multiple orientations that are invariant or quasi-invariant to changes in viewpoint . These features are computed at many salient locations both in the reference and video imagery [8]. The salient locations are determined automatically based on distinctiveness of local image structure. Even with the feature representations

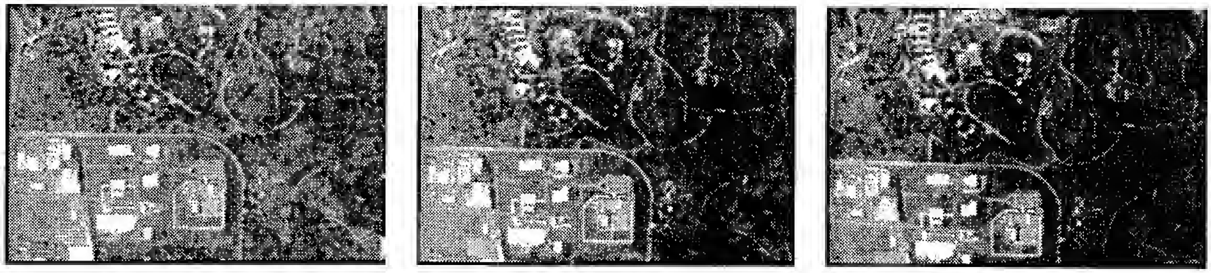


Figure 4: Top: Three frames of an aerial video clip.

at salient locations only, there may be too much data for exhaustive matching. Therefore, in the first step, fast indexing of the multi- dimensional visual features is used to eliminate most of the false matches [2, 12]. Subsequently, exhaustive matching of the small set of remaining candidate matches leads to the correct coarse location of the video imagery in the reference coordinates.

4 Fine Geo-registration

The coarse localization is used to initialize the process of fine alignment while accounting for the geometric and photometric transformations between the video and reference imagery. In general, the transformation between two views of a scene can be modeled by (i) an external coordinate transformation that specifies the 3D alignment parameters between the reference and the camera coordinate systems, and (ii) an internal camera coordinate system to image transformation that typically involves a linear (affine) transformation and non-linear lens distortion parameters. Our approach to the precise alignment problem combines the external coordinate transformation and the linear internal transformation into a single 3D projective view transformation . This along with the depth image and the non-linear distortion parameters completely specifies the alignment transformation between the video pixels and those in the reference imagery. It is to be emphasized that one main advantage of our approach is that no explicit camera calibration parameters need be specified. This aspect tremendously increases the scope of applicability of our proposed system to fairly arbitrary video camera platforms. The modeled video-to-reference transformation is applied to the solution of the precise alignment problem. The process involves simultaneous estimation of the unknown transformation parameters as well as the warped reference imagery that precisely aligns with the video imagery. Multi-resolution coarse-to-fine estimation and warping with Gaussian/Laplacian pyramids is employed.

Once the indexing and registration steps have precisely located a video mosaic in the reference image coordinates, maintenance of the alignment need not be done frequently through indexing. The alignment parameters computed between video frames may be combined with those computed between a video mosaic and the reference to maintain alignment between the video and reference imagery.

4.1 Formulation

We now present the equations used for aligning video imagery to a co-registered reference mage and depth image [2]. The formulation used is the plane+parallax model developed by [6, 11, 13]. The coordinates of a point in a video image are denoted by (x, y). The coordinates of the corresponding point in the reference image are given by (X_r, Y_r). Each point is the reference image has a parallax value k. The parallax value is computed from the dense depth image which is co-registered with the reference image.

There are fifteen parameters $a_1 ... a_{15}$ used to specify the alignment.
The parameters $a_1 .. a_9$ specify the motion of a virtual plane.
The parameters $a_{10} .. a_{12}$ specify the 3D parallax motion.
The parameter a_{13} specifies the lens distortion.
The parameters $a_{14} .. a_{15}$ specify the center for lens distortion.

First the reference image coordinates (X_r, Y_r) are projected to ideal video coordinates (X_I, Y_I):

$$\begin{aligned} X_I &= \frac{(a_1 * X_r + a_2 * Y_r + a_3 + k * a_{10})}{(a_7 * X_r + a_8 * Y_r + a_9 + k * a_{12})} \quad (2) \\ Y_I &= \frac{(a_4 * X_r + a_5 * Y_r + a_6 + k * a_{11})}{(a_7 * X_r + a_8 * Y_r + a_9 + k * a_{12})} \end{aligned}$$

Note, since, the right hand side in the above two equations is a ratio of two expressions, the parameters $a_1 .. a_{12}$ can only be determined up to a scale factor. We typically make parameter $a_9 = 1$ and solve for the remaining 11 parameters.

[2] The equations for aligning video imagery to a co-registered orthophoto and DEM are similar.

The ideal video coordinates (X_I, Y_I) are related to the measured video coordinates (x, y) by the following equation:

$$\begin{aligned} x &= X_I + a_{13} * (X_I - a_{14}) * r^2 & (3) \\ y &= Y_I + a_{13} * (Y_I - a_{15}) * r^2 \\ where & & (4) \\ r^2 &= (X_r - a_{14})^2 + (Y_I - a_{15})^2 \end{aligned}$$

Note, the lens distortion parameters $a_{13}..a_{15}$ may be computed at the video mosaicing stage. In that case, the estimated values are used. However, we have also implemented the above system, where the lens distortion parameters are simultaneously computed with the projective $a_1..a_8$ and epipolar parameters $a_{10}..a_{12}$. The parallax value[3] [6, 11, 13] k at any reference location is calculated from the depth z at that location using the following equation:

$$k = \frac{(z - \bar{z}) * \bar{z}}{z * \sigma_z} \qquad (5)$$

where $\bar{z}$ and σ_z are the average and standard deviation of the depth image values.

4.2 Pre-filtering

The reference imagery and the video are typically acquired at different times. Hence, to correlate the video to the reference imagery, we do the following transformations. We first compute and match the histograms [4] of the video image to the predicted piece of the reference image. This allows us to modify the video image, so that it has a similar histogram as the reference image. Finally, we compute the laplacian pyramids of the reference image and the modified video image. The alignment parameters are computed by correlating these two images.

4.3 Optimization

To register the video image to the reference image, we use a hierarchical direct registration technique [1]. This technique first constructs filter pyramids from each of the two input images, and then estimates the motion parameters in a coarse-fine manner. Within each level the Sum of squared difference (SSD) measure integrated over user selected regions of interest is used as a match measure. This measure is minimized with respect to the unknown transformation parameters $a_1..a_15$. The SSD error measure for estimating the transformation parameters within a region is:

$$E(\{\mathbf{A}\}) = \sum_{\mathbf{x}} (I(\mathbf{x}, t) - I(\mathbf{A}\mathbf{x}), t - 1))^2 \qquad (6)$$

where $\mathbf{x} = (x, y)$ denotes the spatial image position of a point, I the (Laplacian pyramid) image intensity and ($\mathbf{Ax}$ denotes the image transformation at that point (see equations (3) and (4)). The error is computed over all the points within the region. The optimization is done in an iterative manner, at each level of the pyramid using the Levenbreg Marquardt optimization technique.

4.4 Geo-mosaicing, Mapping points, Warping Images

Once, the alignment parameters have been computed, the video images can be warped to the reference image. These video images can then be merged to construct **geo-mosaics** (geo-referenced video mosaics). These mosaics can be used to update the reference imagery. We show examples of the geo-referenced video mosaics constructed using this technique for both nadir and oblique imagery in Figure 8. The original reference image and depth image can be seen in Figure 6. The oblique video image can be seen in Figure 7. The nadir video images can be seen in Figure 4.

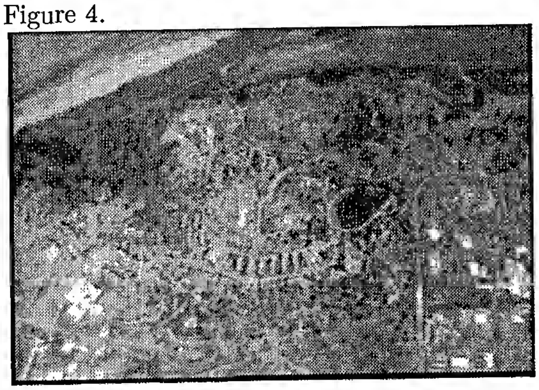


Figure 7: One frame from a video clip captured at a highly oblique angle with respect to the reference imagery. The reference image is the same as in the nadir view case.

Finally, for annotation and other visualization tasks, it is important for the user to be able to map points from the video to the reference image and vice versa. To map points from the reference image to the video, we use equations (3) and (4)) and compute the values on the right hand side. However, to map a video point to the reference image, we solve the equations (3) and (4)) using Newton's method. We use Newton's method in two steps, we first solve equation (4) and then use the results of that to solve equation (3).

Similarly for warping the video image to the reference image, we can use reverse warping with bilinear interpolation. However, to warp the reference image to appear in the video image coordinates, we must use forward warping. Point mappings in the forward warping process are computed using the above technique.

[3] In the case of the reference image being an orthophoto with a corresponding DEM, k is equal to the DEM value

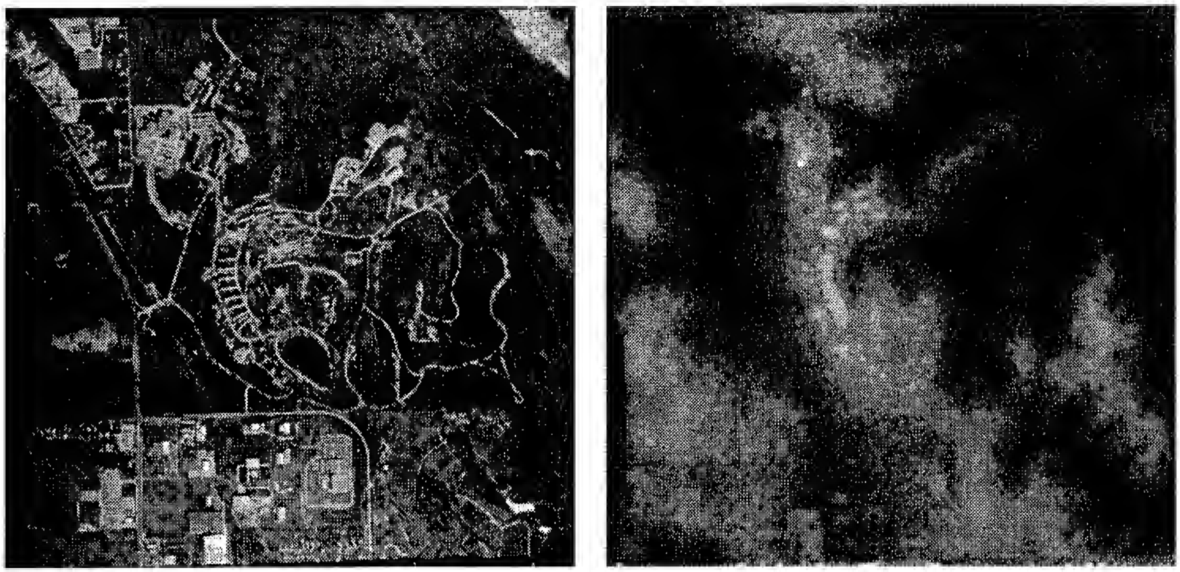


Figure 6: Left: Reference Image, Right: Digital Elevation Map.

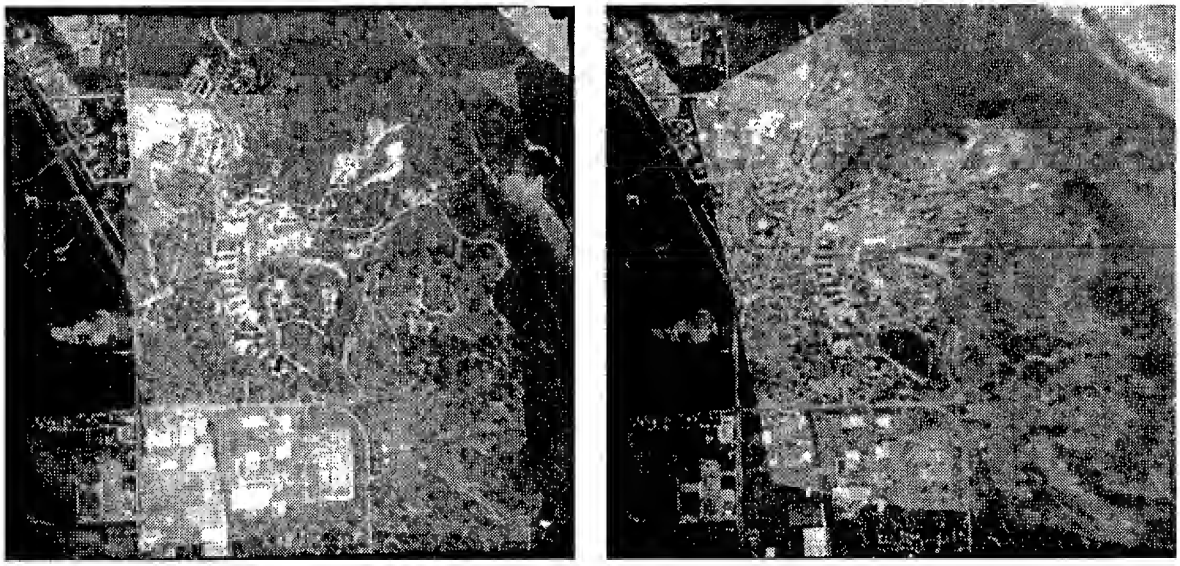


Figure 8: Left: Geo-registered nadir video, and Right: Geo-registered oblique video shown overlaid on the reference image.

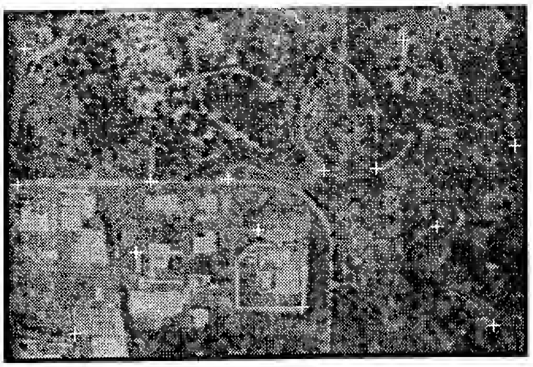


Figure 9: Selected points overlaid on one frame of the aerial video.

In order to assess the approximate accuracy of geo-referencing, a few points were manually selected in a video frame and the corresponding points manually identified in the reference image. Figure 9 shows the selected points marked with +'s overlaid on the video frame. Points in the reference image corresponding to those in the video were also identified using the geo-registration algorithms. Table 1 shows the accuracy of located points in the reference with respect to the hand selected ones. Second and third columns in the table show the coordinates of the selected video points and the subsequent two columns show the corresponding points selected manually in the reference image. The last two columns show the points computed in the reference image by the geo-registration algorithm. Most correspondences are within 1 pixel accuracy with respect to the manually selected locations.

References

[1] J. R. Bcrgcn, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical modcl-based motion cstimation. In *European Conferencc on Computer Vision*, Santa-Margherita Ligurc, Italy, 1992.

[2] P.J. Burt. Smart scnsing with a pyramid vision machinc. *Proceedings of the IEEE*, 76:1006–1015, 1988.

[3] D.S. Linden et. al. Automated digital mosaieking of airborne videography. Technical Report FHTET 96-12, Forest Health Teehnology, Enterprise Team - Fort Collins, Colorado, U.S. Dept. of Agrieulture, Forest Serviee, 1996.

[4] David Hecger and J. R. Bergen. Pyramid-bascd texture analysis and synthesis. In *SIGGRAPH*, 1995.

[5] Michal Irani. Applieations of image mosaies. In *International Conference on Computer Vision*, Cambridge, MA, November 1995.

[6] R. Kumar, P. Anandan, and K. Hanna. Direet recovery of shape from multiple views: a parallax bascd approaeh. In *Proc 12th ICPR*, 1994.

[7] Rakesh Kumar, P.Anandan, J.Bergen, M.Irani, and K.J.Hanna. Seene representation as a eollection of images. In *IEEE Workshop on Visual Representations of Scenes*, Cambridge, MA, June 1995.

Table 1: **Mapping points from video to reference using non-linear technique.** Comparision of automatic estimation vs. hand measurement

Pt. No.	Video Image		Reference Image		Reference Image	
	Input point		Hand Measured		Computed -	
	x pix	y pix	x pix	y pix	x pix	y pix
Points in the center of the image						
1	153	118	280	380	280.26	378.24
2	219	113	341	372	341.65	371.33
3	100	119	231	382	229.90	381.05
4	174	153	300	414	300.84	413.47
5	255	112	376	371	376.27	368.78
6	90	167	221	432	220.72	432.42
Points in the edge of the image						
7	274	23	397	269	397.06	267.75
8	14	26	125	278	124.45	276.11
9	48	222	176	497	175.42	498.84
10	336	220	477	493	477.88	494.06
11	351	97	483	350	481.45	345.84
12	9	120	130	386	132.27	385.72
Other points in the image						
13	204	206	330	469	331.11	468.73
14	297	152	423	411	422.74	411.30
15	119	49	246	309	244.89	306.67

[8] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 1994.

[9] RADIUS PI Reports and Technical papers. Proc. darpa image understanding workshop, 1996. pp. 255–525.

[10] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-based 2D&3D dominant motion estimation for mosaicing and video representation. In *ICCV*, 1995.

[11] Harpreet Sawhney. 3D geometry from planar parallax. In *Proc. CVPR 94*, June 1994.

[12] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 872–877, 1996.

[13] A. Shashua and N. Navab. Relative affine strueture, theory and applieation to 3d reeonstruetion from 2d views. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[14] R. Szeliski. Image mosaieing for tele-reality applieations. In *IEEE Wkshp. on Applications of Computer Vision*, pages 44-53, 1994.

5 Acknowledgement

The reference image and DEM shown were provided to us by GDE Systems Inc.. They were produced using the GDE SocetSet photogrammetry system. We thank Steve Hsu of Sarnoff for his comments on the paper.

Matching and Pose Refinement with Camera Pose Estimates

Satyan Coorg* **Seth Teller**
MIT Computer Graphics Group
545 Technology Square NE43-217
Cambridge MA 02139
satyan@graphics.lcs.mit.edu, http://graphics.lcs.mit.edu

Abstract

This paper describes novel algorithms that use absolute camera pose information to identify correspondence among point features in hundreds or thousands of images. Our *incidence counting* algorithm is a *geometric* approach to matching; it matches features by extruding them into an absolute 3-D coordinate system, then searching 3-D space for regions into which many features project.

The absolute pose estimates reported by our instrumentation are accurate, but not perfect. Thus, we also consider the problem of *refining* these pose estimates, given feature matches from a set of images. We describe a pose refinement algorithm which decouples translation (position) estimates from rotation (attitude) estimates, and can incorporate matches from many hundreds or thousands of images.

1 Introduction

Many 3-D reconstruction algorithms rely on a *matching* or *correspondence* step to identify constraints corresponding to the scene geometry; these constraints are used to guide the 3-D reconstruction process. Typically, matching is performed using some image attribute (e.g., pixel luminance [Gennery, 1977]) or some geometric attribute (e.g., length and orientation of edges [Ayache, 1991]). While these techniques work well for images taken from nearby camera positions, they are less effective for disparate images taken from cameras that are far from each other.

In this paper, we design a matching algorithm that uses camera pose estimates (provided by physical instrumentation) to *over-constrain* the matching problem, identifying matches by applying geometric constraints imposed by the camera positions. In some ways, our algorithm is similar to use of the *epipolar* constraint in stereo vision [Faugeras, 1993], but generalizes that method in its incorporation of many cameras and images.

As the absolute pose estimates reported by our instrumentation are not perfect, we also consider the problem of camera pose refinement, i.e., computing accurate camera poses for many images, given matches between points and fairly accurate initial pose estimates.

Much of the existing research on pose refinement has revolved around the assumption that *no* 3-D information is available [Mohr *et al.*, 1995, Faugeras, 1992]. The basis of these algorithms is the epipolar constraint between two images:

$$\tilde{\mathbf{m}}^T \mathbf{F} \tilde{\mathbf{m}}' = 0$$

where $\mathbf{F}$ is the 3 × 3 *fundamental* matrix relating two (projective) points $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{m}}'$ in the two images. Determining the fundamental matrix is equivalent to determining the (relative) poses of the two cameras involved. Given eight

*Funding for this research was provided in part by the Advanced Research Projects Agency of the Department of the Defense under Office of Naval Research contract N00014-92-J-1310.

or more point correspondences, it is possible to determine the fundamental matrix up to a scale factor using the *eight point* algorithm [Longuet-Higgins, 1981]. However, typical algorithms [Faugeras, 1992, Hartley, 1995] use more points than eight in order increase the robustness of the algorithm.

While this technique performs well for pairs of images, there are several disadvantages in using the fundamental matrix technique for a large number of images. First, these algorithms involve only pairwise matching; using them to compute pose for m cameras pairwise may result in large "drift" error. Second, they determine camera pose and 3-D positions only up to a projective transformation, which needs to be "corrected" as a post-processing step. Third, the use of projective matrices increases the complexity of the solution because of greater number of variables and more complicated constraints (such as singularity).

In contrast, we formulate the problem as a direct 3-D optimization algorithm that *refines* initial camera pose estimates. One advantage of this approach is that the number of unknown variables is less, increasing the robustness of the algorithm. Also, the algorithm can seamlessly incorporate matches across many images. Finally, from a practical standpoint, it is much easier to visualize and debug (using computer graphics) algorithms operating in 3-D; this would be much harder for algorithms that operate in more complex spaces.

2 Incidence Counting

The incidence counting algorithm is based on the following property of projection: if any *sparse* set of features in multiple images are extruded to 3-D, then it is likely that regions of high incidence (regions where extrusions from multiple cameras intersect) correspond to real 3-D features. Figure 1 illustrates the idea of the algorithm in 2-D. In the figure, $E1$, $E2$, and $E3$ are cameras imaging three features (points A, B, C). The extrusions of the image features are *rays* originating from the camera and passing through the feature. If a feature is present in k images, k rays would intersect at the location corresponding to the feature (e.g., points A, B, C all have high incidence of $k = 3$). Thus, a simple way to identify matches would be search for regions with high incidence; an efficient method to perform the search is described in Section 2.1.

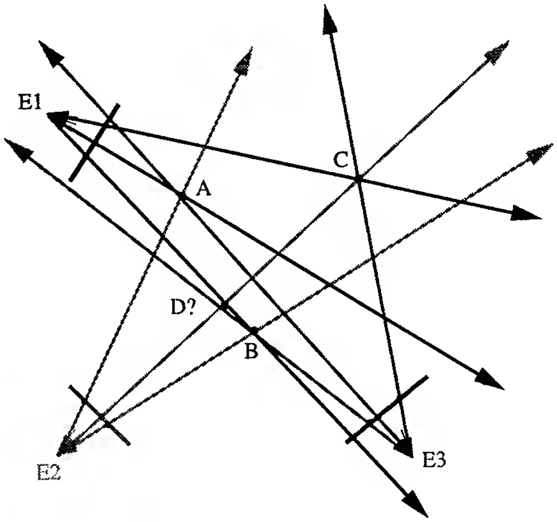


Figure 1: Incidence counting in two dimensions.

Note that, in addition to the "true" features, there are also *spurious* regions with high incidence. For example, even though point D was not one of the features imaged by the cameras, D has the property that rays from all three cameras pass close to it; i.e., D is a possible candidate for a match. Section 2.2 provides a method to eliminate some spurious matches by associating an error value with each 3-D position. Future work will incorporate methods using image attributes (color and texture) to eliminate additional spurious matches.

2.1 Octree-Based Incidence Counting

Our algorithm for incidence counting requires two parameters in addition to the images and camera poses:

- ϵ, a "nearness" threshold. This is necessary to handle (small) errors in either the location of the image feature or in camera pose. The choice of ϵ depends on both the accuracy of the camera pose estimate, and the desired accuracy of the reconstruction.

- k, the incidence threshold. It is related to the *density* of camera positions relative to the features of interest – a reasonable value would be the average number of cameras imaging a feature.

Given these parameters, points of high incidence are those for which k or more rays pass by within a distance ϵ.

Possible methods of identifying high incidence are to check the above condition for (1) all k-cardinal subsets of the set of rays, or (2) all 3-D points in a discrete set (e.g., regular grid). Both these methods have disadvantages. Checking all possible subsets suffers from a combinatorial increase in complexity with k; checking only a discrete set of 3-D points suffers from the usual problems of point sampling (i.e., missing some 3-D feature (undersampling), or inefficiency (oversampling)).

Fortunately, rays constructed by extrusion exhibit the clustering property: while there are regions of high density (e.g., near features), there are large regions containing very few rays. We exploit this property by constructing an octree [Samet, 1990] to store the rays. The octree is constructed by associating the region of interest (a bounding-box overestimate of the cameras and the scene to be modeled) with the root node, and subdividing octree nodes until either each leaf node is associated with fewer than k rays[1], or its dimensions are less than ϵ. Once the octree has been constructed, each leaf node is examined to check whether the rays through it pass through within ϵ of each other. This can be performed by computing the (least-squares) best point lying on all these rays. The algorithm reports all the points (and corresponding rays) whose error is less than ϵ.

2.2 Eliminating Spurious Matches

As mentioned earlier, one drawback of the incidence counting algorithm is that it identifies even spurious matches. In this section, we design an algorithm to eliminate some spurious matches by enforcing the constraint that a single ray can contribute to at most one 3-D point. The algorithm given below uses the error metric associated with a 3-D point to choose at most one 3-D point for each ray. Informally, it uses the criteria that 3-D points with low error are retained, and those with high error are rejected.

Algorithm Check-Spurious:

```
1. Sort all (say, $n$) high incidence 3-D points according to their error (the lowest error being first). $\mathbf{P_i}$ denotes the $i^{th}$ 3-D point.
2. foreach $1 \leq i \leq n$ do
    (a) if ($\mathbf{P_i}$ is invalid) continue;
    (b) Output $\mathbf{P_i}$ as a valid point.
    (c) foreach $i < j \leq n$ such that $\mathbf{P_i}$ and $\mathbf{P_j}$ share a ray, mark $\mathbf{P_j}$ as invalid.
```

This algorithm also has the property that it computes the minimum error valid configuration (in a lexicographic sense).

3 A Direct 3-D algorithm for Camera Pose Refinement

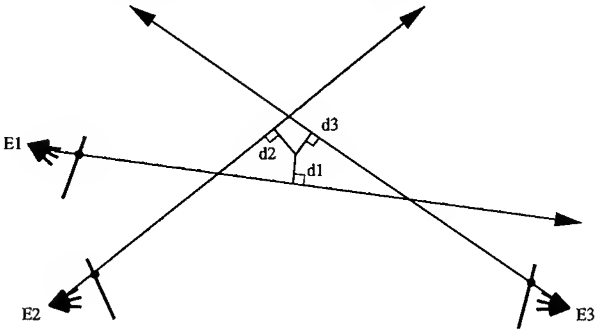


Figure 2: Reconstruction using least squares of distances.

We now present an algorithm for refining camera pose estimates, given matches across points in different images. Figure 2 illustrates (in 2-D) the idea behind our algorithm. If the camera poses are accurate, then the rays constructed by extrusion would pass through the reconstructed 3-D point. Typically, due to error in the cam-

[1] We associate a ray with an octree node if it intersects the ϵ-extended box around the node.

era pose estimates, they will diverge from the reconstructed point. This can be used to "correct" the camera poses so that the rays are as close as possible to the 3-D reconstruction.

Formally, the pose refinement problem is as follows. Given:

- For $1 \leq i \leq m$, $\mathbf{E'}_i$ and R'_i – the translation and rotation estimates of the i^{th} camera;

- For $1 \leq i \leq m$, $1 \leq j \leq n$, rays $\mathbf{v}_{ij}$ – unit vectors that correspond to projections of point $\mathbf{P}_j$ from camera i (in the camera's coordinate system);

compute $\mathbf{E}_i$, R_i for $1 \leq i \leq m$ (the true pose of each camera), and $\mathbf{P}_j$ for $1 \leq j \leq n$ (the correct 3-D positions each matched point).

We formulate the problem as a minimization of the following objective function[2]:

$$O = \sum_{i=1}^{m}\sum_{j=1}^{n} \|(\mathbf{P}_j - \mathbf{E}_i) \times R_i(\mathbf{v}_{ij})\|^2$$

Geometrically, this function represents the sums of the squared distances from reconstructed points to their corresponding rays (Figure 2).

As the objective function does not have a linear least-squares form, we use an iterative method to solve for camera pose. Our approach is to consider the problems of finding each transformation independently (assuming the other is known accurately) and combining the two methods when neither translations nor rotations are known exactly. While this is equivalent to minimizing the objective function using partial derivatives with respect to translations and rotations, it is helpful to separate the two cases for clearer presentation; solutions to these two cases turn out to be quite different.

3.1 Translations

In this section, we solve for translations of the cameras, assuming that their rotations are known accurately. Thus, $R_i(\mathbf{v}_{ij})$ can be replaced by a (known) unit vector $\mathbf{v'}_{ij}$. The resulting objective function has the following form:

$$O = \sum_{i=1}^{m}\sum_{j=1}^{n} \|(\mathbf{P}_j - \mathbf{E}_i) \times \mathbf{v'}_{ij}\|^2$$

which can be written as:

$$O = \sum_{i=1}^{m}\sum_{j=1}^{n} \|\mathbf{L'}_{ij}(\mathbf{P}_j - \mathbf{E}_i)\|^2$$

where $\mathbf{L'}_{ij}$ is the 3×3 skew-symmetric matrix defining the cross product whose elements are determined by the components of $\mathbf{v'}_{ij}$:

$$\begin{bmatrix} 0 & v'_{ij,3} & -v'_{ij,2} \\ -v'_{ij,3} & 0 & v'_{ij,1} \\ v'_{ij,2} & -v'_{ij,1} & 0 \end{bmatrix}$$

Writing $\|\mathbf{x}\|^2 = \mathbf{x}.\mathbf{x}$ as $\mathbf{x}^T\mathbf{x}$, we obtain:

$$O = \sum_{i=1}^{m}\sum_{j=1}^{n} (\mathbf{P}_j - \mathbf{E}_i)^T \mathbf{L'}^T_{ij}\mathbf{L'}_{ij}(\mathbf{P}_j - \mathbf{E}_i)$$

This is of the form $\mathbf{x}^T\mathbf{A}\mathbf{x}$ for where $\mathbf{A}$ is a symmetric matrix. The derivative of this function with respect to $\mathbf{x}$ is $\mathbf{A}\mathbf{x}$.

Computing the derivatives of this function with respect to $\mathbf{P}_j$, and setting it to $\mathbf{0}$ yields:

$$\sum_{i=1}^{m} \mathbf{L'}^T_{ij}\mathbf{L'}_{ij}(\mathbf{P}_j - \mathbf{E}_i) = \mathbf{0}$$

Thus, $\mathbf{P}_j = \mathbf{A}^{-1}\mathbf{b}$, where

$$\mathbf{A} = \sum_{i=1}^{m} \mathbf{L'}^T_{ij}\mathbf{L'}_{ij}$$

$$\mathbf{b} = \sum_{i=1}^{m} \mathbf{L'}^T_{ij}\mathbf{L'}_{ij}\mathbf{E}_i$$

Geometrically, this solution gives the point that minimizes the sum of squared distances of $\mathbf{P}_j$ from the corresponding rays.

[2] Note that $\mathbf{P}_j = \mathbf{0}$ and $\mathbf{E}_i = \mathbf{0}$ is a trivial solution to the minimization problem. This can be avoided by imposing a constraint that the sum of their magnitudes must be some non-zero constant. In practice, due to the use of initial pose estimates, we have found that the optimization converges to non-trivial solutions.

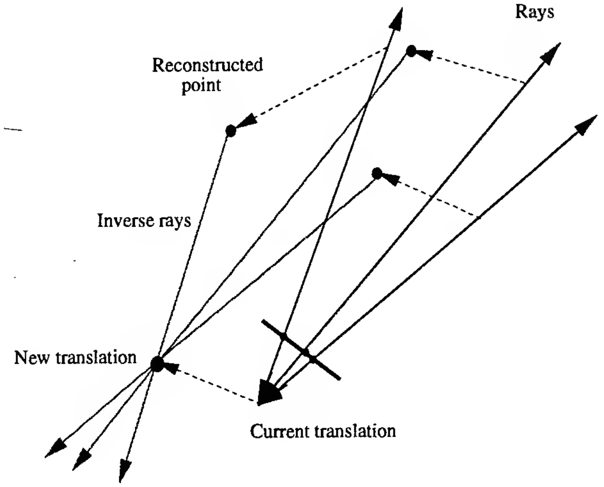


Figure 3: Translation estimate using inverse rays.

As the objective function is symmetrical in $\mathbf{P}_j$ and $\mathbf{E}_i$, setting the derivative with respect to $\mathbf{E}_i$ yields the equation $\mathbf{E}_i = \mathbf{A}^{-1}\mathbf{c}$, where

$$\mathbf{c} = \sum_{j=1}^{n} \mathbf{L'}^T_{ij}\mathbf{L'}_{ij}\mathbf{P}_j$$

This is equivalent to finding the 3-D point that minimizes the sum-of-squared distances from the "inverse" rays through $\mathbf{P}_1 \ldots \mathbf{P}_n$ (Figure 3).

The translation refinement algorithm alternately computes 3-D positions and camera translation estimates using the equations given above[3]. Convergence in the algorithm is detected by little change in the objective function.

3.2 Rotations

The first step in an optimization involving unknown rotations is to choose a representation for expressing rotations. A variety of representations are in use: orthonormal matrices, quaternions, Euler angles, etc. [Foley *et al.*, 1990]. Each of these representations has its own advantages and disadvantages; the most appropriate representation depends on the application (e.g., quaternions provide closed form solutions for absolute orientation [Horn, 1987]). For this optimization, we chose to use Euler angles, i.e., rotation is represented by three rotations about the coordinate axes. This has the advantage that no additional constraints are needed to ensure rotational properties, in contrast to the orthonormality constraint for 3 × 3 matrices or the unit length constraint for quaternions. This allows use of simple (unconstrained) non-linear optimization methods such as the Newton-Raphson method [Scales, 1985] to solve for the rotation parameters.

Rotations are represented as:

$$\mathbf{R}^x(r_i)\mathbf{R}^y(s_i)\mathbf{R}^z(t_i)$$

where r_i, s_i, t_i are the Euler angles, and $\mathbf{R}^{\{x,y,z\}}$ are 3×3 matrices representing rotations about the coordinate axis. For example, $\mathbf{R}^z(\theta)$ is the matrix:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We use the iterative Newton-Raphson method using the gradient (a vector formed by the first partial derivatives) and Hessian (a matrix formed by the second partial derivatives) of the objective function to solve for the camera rotations [Scales, 1985]. Given initial estimates of r, s, t for some camera i (subscripts are omitted for clarity), increments $\Delta r, \Delta s, \Delta t$ are defined by the gradient and the Hessian:

$$\begin{bmatrix} \frac{\partial^2 O}{\partial r^2} & \frac{\partial^2 O}{\partial r \partial s} & \frac{\partial^2 O}{\partial r \partial t} \\ \frac{\partial^2 O}{\partial r \partial s} & \frac{\partial^2 O}{\partial s^2} & \frac{\partial^2 O}{\partial s \partial t} \\ \frac{\partial^2 O}{\partial r \partial t} & \frac{\partial^2 O}{\partial s \partial t} & \frac{\partial^2 O}{\partial t^2} \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta s \\ \Delta t \end{bmatrix} = \begin{bmatrix} -\frac{\partial O}{\partial r} \\ -\frac{\partial O}{\partial s} \\ -\frac{\partial O}{\partial t} \end{bmatrix}$$

The partial derivatives are obtained by symbolically differentiating the objective function with respect to r, s, t and evaluating the expressions using the current values of r, s, t. Some of the partial derivative expressions are listed in the appendix.

Given the current rotation in terms of r, s, t, the rotation refinement algorithm evaluates the partial derivative expressions and computes $\Delta r, \Delta s, \Delta t$. The new rotations are used to update the 3-D positions of the reconstructed points, and this process is repeated until convergence.

[3] The solution is valid only up to a rigid (rotation, translation, uniform scaling) transformation. The "correct" transformation can be obtained by fixing the values of some three points in absolute coordinates.

4 Conclusion

We presented the incidence counting algorithm that identifies matches using only the geometric constraints implied by camera pose. The algorithm performs fairly well for synthetic images and camera pose [Coorg and Teller, 1996], but more experiments on real data are needed to fully evaluate its efficacy.

We also presented a direct 3-D algorithm to refine camera pose estimates given correspondences. Our algorithm operates directly in 3-D and can easily incorporate matches across hundreds or thousands of images. Results of this algorithm on synthetic data (random 3-D points, perturbed camera poses) is presented in [Coorg and Teller, 1996]; we plan to experiment with real data when our pose-instrumented platform is operational.

References

[Ayache, 1991] Nicholas Ayache. *Artificial Vision for Mobile Robots.* The MIT Press, Cambridge, MA, 1991.

[Coorg and Teller, 1996] Satyan Coorg and Seth Teller. Matching and pose refinement with camera pose estimates. Technical Report TM-561, Laboratory for Computer Science, MIT, 1996.

[Faugeras, 1992] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In Giulio Sandini, editor, *Proceedings of Computer Vision (ECCV '92)*, volume 588 of *LNCS*, pages 563–578, Berlin, Germany, mai 1992. Springer.

[Faugeras, 1993] Olivier Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[Foley *et al.*, 1990] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, Reading, Massachusetts, 1990.

[Gennery, 1977] D. B. Gennery. A stereo vision system for an autonomous vehicle. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 576–582, August 1977.

[Hartley, 1995] R. Hartley. In defence of the 8-point algorithm. In *ICCV95*, pages 1064–1070, 1995.

[Horn, 1987] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4), April 1987.

[Longuet-Higgins, 1981] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[Mohr *et al.*, 1995] R. Mohr, L. Quan, and F. Veillon. Relative 3d reconstruction using multiple uncalibrated images. *IJRR*, 14(6):619–632, December 1995.

[Samet, 1990] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS.* Addison-Wesley, 1990.

[Scales, 1985] L.E. Scales. *Introduction to Non-Linear Optimization.* Springer-Verlag, 1985.

A Rotational Partial Derivatives

We only list the partial derivatives with respect to t; the expressions for r and s are similar. Let,

$$\begin{aligned}
\mathbf{S}^z(\theta) &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\mathbf{D}_j &= \mathbf{P}_j - \mathbf{E} \\
\mathbf{V}_j &= \mathbf{D}_j \times (\mathbf{R}^x(r)\mathbf{R}^y(s)\mathbf{R}^z(t)\mathbf{v}_{ij}) \\
\mathbf{V}_j^t &= \mathbf{D}_j \times (\mathbf{R}^x(r)\mathbf{R}^y(s)\mathbf{S}^z(t+\frac{\pi}{2})\mathbf{v}_{ij}) \\
\mathbf{V}_j^{rt} &= \mathbf{D}_j \times (\mathbf{S}^x(r+\frac{\pi}{2})\mathbf{R}^y(s)\mathbf{S}^z(t+\frac{\pi}{2})\mathbf{v}_{ij}) \\
\mathbf{V}_j^{tt} &= \mathbf{D}_j \times (\mathbf{R}^x(r)\mathbf{R}^y(s)\mathbf{S}^z(t+\pi)\mathbf{v}_{ij})
\end{aligned}$$

Then,

$$\begin{aligned}
\frac{\partial O}{\partial t} &= \sum_{j=1}^{n} \mathbf{V}_j.\mathbf{V}_j^t \\
\frac{\partial^2 O}{\partial t^2} &= 2\sum_{j=1}^{n} \mathbf{V}_j^t.\mathbf{V}_j^t + 2\sum_{j=1}^{n} \mathbf{V}_j.\mathbf{V}_j^{tt}
\end{aligned}$$

Unifying Two-View and Three-View Geometry

Shai Avidan Amnon Shashua *
Institute of Computer Science,
The Hebrew University,
Jerusalem 91904, Israel
e-mail: avidan@cs.huji.ac.il

Abstract

The core of multiple-view geometry is governed by the fundamental matrix and the trilinear tensor. In this paper we unify both representations by first deriving the fundamental matrix as a rank-2 trivalent tensor, and secondly by deriving a unified set of operators that are transparent to the number of views. As a result, we show that the basic building block of the geometry of multiple views is a trivalent tensor that specializes to the fundamental matrix in the case of two views, and is the trilinear tensor (rank-4 trivalent tensor) in case of three views. The properties of the tensor (geometric interpretation, contraction properties, etc.) are independent of the number of views (two or three). As a byproduct, every two-view algorithm can be considered as a degenerate three-view algorithm and three-view algorithms can work with either two or three images, all using one standard set of tensor operations. To highlight the usefulness of this paradigm we developed two applications. A novel view synthesis algorithm that starts with the rank-2 tensor and seamlessly move to the general rank-4 trilinear tensor, all using one set of tensor operations. We also applied it to a camera stabilization algorithm.

1 Introduction

The geometry of multiple views is governed by certain multi-linear constraints, bilinear for pairs of views and trilinear for triplets of views — all other multilinear constraints (four views and beyond) are spanned by the bilinear and trilinear constraints. The bilinear constraint determines the "fundamental matrix" and the trilinear constraints determine the "trilinear tensor". The fundamental matrix is a rank-2 3×3 matrix and the trilinear tensor is a rank-4 trivalent tensor. There are known properties of the fundamental matrix, there are known properties of the trilinear tensor, and there are known connections between the two — for instance how to extract the fundamental matrix from the trilinear tensor. There are algorithms (for reconstruction, view synthesis, camera stabilization) that are defined for concatenation of pairs of views, and there are algorithms that are defined for concatenation of triplets of views. What is needed, therefore, is a canonical representation, a single object with a standard set of operators, that applies uniformly to pairs or triplets of views. In other words, the unification efforts that have appeared so far in the literature focus on the transformation groups (projective, affine and Euclidean) represented by the camera matrix, leading to a canonical framework [3, 10, 6] for the geometry of two views. Given the recent progress on multi-linear tensorial constraints across more than two views, there is a need to make a similar unification attempt but now across the temporal axis (number of views), rather than on the spatial axis (transformation groups).

The paper has two main results. First, we establish a set of operators that are used to synthesize tensors from one another. Second, we derive the geometry of two views using those operators and show that the familiar fundamental matrix is embedded in a rank-2 trivalent tensor (of 27 coefficients). We show that the properties of the rank-2 tensor are identical with the known properties of the rank-4 trilinear tensor (of three distinct views), and the set of operators apply uniformly to both tensors. As a result, the geometry of multiple views is governed by a single tensorial structure with a standard set of operators and is uniform with respect to the number of views — the only change that occurs when the number of views is two is that the rank of the tensor becomes 2 instead of 4, but this does not have an effect on the manner in which the tensor is used for applications.

Apart from the theoretical result, we show practi-

*This work was partly funded by ARPA through the U.S. Office of Naval Research under grant - N00014-93-1-1202, R&T Project Code 4424341—01, US-Israel binational science foundation 94-00120, and the European ACTS project AC074 "VANGUARD"

cal benefits of this unification step. First is the "cross-platform" capability of algorithms to work both in the case of two and three views, as the latter is simply a generalization of the former. This results in the ability to handle freely and seamlessly the geometry of two and three images in a single framework. Instead of existing two-views algorithms one can use three-view based algorithms, taking advantage of the third view, in case it is present, but working with two images as well without modification, all due to the introduction of the rank-2 tensor. To demonstrate these properties we present two applications — a novel view synthesis algorithm that highlights the simple handling of the geometry of two or three images and a video stabilization algorithm that works, as is, with two or three images.

2 Background and Notations

We assume that the physical 3D world is represented by the 3D projective space $\mathcal{P}^3$ (object space) and its projections onto the 2D projective space $\mathcal{P}^2$ defines the image space. If $x \in \mathcal{P}^3$ varies over the object space, represented by a tetrad of homogeneous coordinates, and $p \in \mathcal{P}^2$ is its projection (represented by a triplet of coordinates), then there exists a 3×4 matrix A satisfying the relation $p \cong Ax$, where $\cong$ represents equality up to scale and A is called the camera matrix. Since only relative camera positioning can be recovered from image measurements, the first camera matrix can be represented by $[I; 0]$.

We will occasionally use tensorial notations, which are briefly described next. We use the covariant-contravariant summation convention: a point is an object whose coordinates are specified with superscripts, i.e., $p^i = (p^1, p^2, ...)$. These are called contravariant vectors. An element in the dual space (representing hyper-planes — lines in $\mathcal{P}^2$), is called a covariant vector and is represented by subscripts, i.e., $s_j = (s_1, s_2, ...)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^i s_i = p^1 s_1 + p^2 s_2 + ... + p^n s_n$. This is known as a contraction. Vectors are also called 1-valence tensors. 2-valence tensors (matrices) have two indices and the transformation they represent depends on the covariant-contravariant positioning of the indices. When viewed as a matrix the row and column positions are determined accordingly: in a_i^j and a_{ji} the index i runs over the columns and j runs over the rows, thus $b_j^k a_i^j = c_i^k$ is $BA = C$ in matrix form. An outer-product of two 1-valence tensors (vectors), $a_i b^j$, is a 2-valence tensor c_i^j whose i, j entries are $a_i b^j$ — note that in matrix form $C = ba^\top$. An n-valence tensor described as an outer-product of n vectors is a rank-1 tensor. Any n-valence tensor can be described as a sum of rank-1 n-valence tensors. The rank of an n-valence tensor is the *smallest* number of rank-1 n-valence tensors with sum equal to the tensor. For example, a rank-1 trivalent tensor is $a_i b_j c_k$ where a_i,b_j and c_k are three vectors. The rank of a trivalent tensor α_{ijk} is the smallest r such that,

$$\alpha_{ijk} = \sum_{s=1}^{r} a_{is} b_{js} c_{ks}. \tag{1}$$

The tensor of vector products is denoted by ϵ_{ijk} (indices range 1-3) operates on two contravariant vectors of the 2D projective plane and produces a covariant vector in the dual space (a line): $\epsilon_{ijk} p^i q^j = s_k$, which in vector form is $s = p \times q$, i.e., s is the vector product of the points p and q.

Two views $p = [I; 0]x$ and $p' \cong Ax$ are known to produce a bilinear matching constraint whose coefficients are arranged in a 3×3 matrix F known as the "Fundamental matrix" of [2] described in the setting of projective geometry (uncalibrated cameras):

$$f_{ij} = \epsilon_{ikl} v'^k a_j^l \tag{2}$$

where $A = [a; v']$ (a_j^l is the left 3×3 minor of A, and v' is the fourth column, the epipole, of A). The bilinear constraint is $f_{ij} p^i p'^j = 0$.

Three views, $p = [I; 0]x, p' \cong Ax$ and $p'' \cong Bx$, are known to produce four trilinear forms whose coefficients are arranged in a tensor representing a bilinear function of the camera matrices A, B:

$$\alpha_i^{jk} = v'^j b_i^k - v''^k a_i^j \tag{3}$$

where $B = [b, v'']$. The four trilinear constraints are:

$$p^i s_j^\mu r_k^\rho \alpha_i^{jk} = 0 \tag{4}$$

where s_j^μ are any two lines (s_j^1 and s_j^2) intersecting at p', and r_k^ρ are any two lines intersecting p''. Since the free indecis are μ, ρ each in the range 1,2, we have 4 trilinear equations (which are unique up to linear combinations). By changing the order of the views one can obtain at most 12 trilinear constraints arranged in three such tensors. These constraints first became prominent in [8] and the underlying theory has been studied intensively since in [11, 5, 4].

The elements of α_i^{jk} satisfy certain properties. The algebraic relations among the elements are described in [4], and contraction properties in [11]. Among the contraction properties it will be useful for later to mention that $\delta_k \alpha_i^{jk}$ (for any vector δ) produces a 2D projective transformation (a homography) from image 1 to 2 via some plane of reference, and $\eta_j \alpha_i^{jk}$ produces a homography matrix from image 1 to 3 via some plane. This homography matrices can be used to recover the fundamental matrix or the epipole. Finally, it has been recently shown in [9] that the rank of α_i^{jk} is 4 (we will return to tensor ranks later).

3 The Basic Tensorial Operators

The basic operation described next describes the transformation a tensor of three views undergoes when one of the cameras changes its position. In other words, this operator can be used to create a chain of tensors, each created from its predecessor in the chain. Later, we will start from a Null tensor (all three views are repeated) and create a chain that along the way creates a representation of two views as a rank-2 trivalent tensor.

Theorem 1 *Given a tensor α_i^{jk} of camera positions $[I;0], A, B$, then the tensor γ_i^{jk} of camera positions $[I;0], A, C$, where C is obtained from B by an incremental change of coordinates R and translation t from the position of the third camera has the form:*

$$\boxed{\gamma_i^{jk} = r_l^k \alpha_i^{jl} - t^k a_i^j} \tag{5}$$

Proof: The proof is beyond the scope of this paper. Please refer to [1]. □

Likewise, if we apply an incremental change to the position of the second camera, rather than the third, then the tensor γ_i^{jk} will have the form:

$$\boxed{\gamma_i^{jk} = r_l^j \alpha_i^{lk} + t^j b_i^k} \tag{6}$$

4 The rank-2 Trivalent Tensor of Two Views

We will use the tensorial operators (eqns. 5 and 6) to create a chain starting from the (Null) tensor of views $< 1,1,1 >$ (all three views are repeated), to tensor of views $< 1,2,1 >$, to tensor $< 1,2,2 >$ and finally to tensor α_i^{jk} of views $< 1,2,3 >$. All the tensors of the chain are trivalent tensors, and of interest are the tensors that represent only two distinct views.

The tensor β_i^{jk} of views $< 1,2,1 >$ can be derived from the Null tensor using eqn. 6,

$$\begin{aligned} \beta_i^{jk} &= a_l^j \alpha_i^{lk} + v'^j b_i^k \\ &= a_l^j 0_i^{lk} + v'^j I_i^k \\ &= v'^j I_i^k, \end{aligned} \tag{7}$$

by the incremental motion $A = [a, v']$ from views $< 1,1,1 >$ to views $< 1,2,1 >$. The elements of the tensor are either 0 or the epipole v'.

Next, we apply an incremental motion of the the third view going from tensor of views $< 1,2,1 >$ to tensor of views $< 1,2,2 >$. The incremental motion is again $A = [a, v']$ and we use the operator described in eqn 5 to obtain:

$$\begin{aligned} \gamma_i^{jk} &= a_l^k \beta_i^{jl} - v'^k a_i^j \\ &= a_l^k (v'^j I_i^l) - v'^k a_i^j \\ &= v'^j a_i^k - v'^k a_i^j \end{aligned} \tag{8}$$

and γ_i^{jk} is the tensor of the image triplet $< 1,2,2 >$. It can be readily verified that the elements of γ_i^{jk} are composed of the fundamental matrix $f_{ij} = \epsilon_{ikl} v'^k a_j^l$, $-f_{ij}$, and the remaining (nine) elements vanish. In other words, we have derived a trivalent tensor representing the geometry of two views, and is composed of the elements of the fundamental matrix. The following theorems and corollary are proved in [1].

Theorem 2 *The rank of the trivalent tensor of two views,*

$$\gamma_i^{jk} = v'^j a_i^k - v'^k a_i^j$$

is 2.

Theorem 3 *the tensor γ_i^{jk} shares the same properties as the general rank-4 tensor of three views.*

A byproduct of these properties is that we can characterize the family of rank-2 homography matrices:

Corollary 1 $[c]_\times F$, *where $c = (c_1, c_2, c_3)$ is a general 3-vector, defines a family of homography matrices from the first image to the second image due to a plane passing through the center of projection of the second camera.*

The corollary extends the result of [6] that $[v']_\times F$ is a homography matrix to a family of homography matrices $[c]_\times F$ passing thru the center of projection of the second camera.

Finally, note that the bilinear constraint follows from γ_i^{jk} in the same manner as in the general rank-4 tensor: $p^i s_j r_k \gamma_i^{jk} = 0$ describes a contraction with the point p in the first view, some line s passing through p' and some other line r passing through p' as well. Thus, we get the same point-line-line interpretation we get with the general rank-4 tensor.

The last tensor in the chain is to go from tensor of views $< 1,2,2 >$ to the general tensor of views $< 1,2,3 >$. This can be readily done using the operator of eqn. 5.

To conclude, we have shown the basic "building block" of stereo vision to be the trilinear tensor of three cameras. Every other object, be it the epipole or the fundamental matrix, is merely a degenerate case of the general trilinear tensor. Since camera parameters can be recovered directly from the trilinear tensor, there is no need for the fundamental matrix, other than to serve as a tool for constructing the rank-2 trivalent tensor in case only two, rather than three, views are given. As a result algorithms developed under the three-view paradigm will apply to all camera configurations, be it two or three cameras.

5 Applications

This section presents two applications to highlight two of the ideas advocated in this paper. The first example highlights the simple and uniform way to treat tensors (both rank-4 and rank-2) in order to obtain new ones. There is no need to distinguish between the geometry of two and three views. Specifically we present an image-based rendering algorithm that starts with a pair of images, related by a rank-2 tensor, and generate a novel view by seamlessly moving from the rank-2 tensor to the rank-4 tensor. The second application demonstrates the generality of algorithms developed in tensor context - they act the same both for the case of two views and three views. We show this on a stabilization algorithm originally developed in the three-view framework.

5.1 Novel View Synthesis

Novel view synthesis, also referred to as image-based rendering, aims at synthesizing novel views of a scene from a given pair of images, without first reconstructing the 3D model. This method can be faster and more accurate to compute than building the 3D model first. The trilinear tensor is an ideal candidate for image-based rendering system, as it is numerically stable and has no degenerate configurations. We use the basic tensor operators described earlier and the rank-2 trivalent tensor of two views to build new tensors. Once a tensor is built we use equation 4 to reproject the novel image.

5.2 Video Stabilization

This application illustrates the "cross-platform" capability of three-view algorithms. As an example we show how to convert a three-view stabilization algorithm, originally presented in [7], to work with two images only. The purpose of the stabilization algorithm was defined to cancel rotation between successive frames. The original paper makes use of the fact that the tensor is composed of three homography matrices to establish a linear relation between the elements of the trilinear tensor to those of the rotation matrix, in case the cameras are calibrated and the

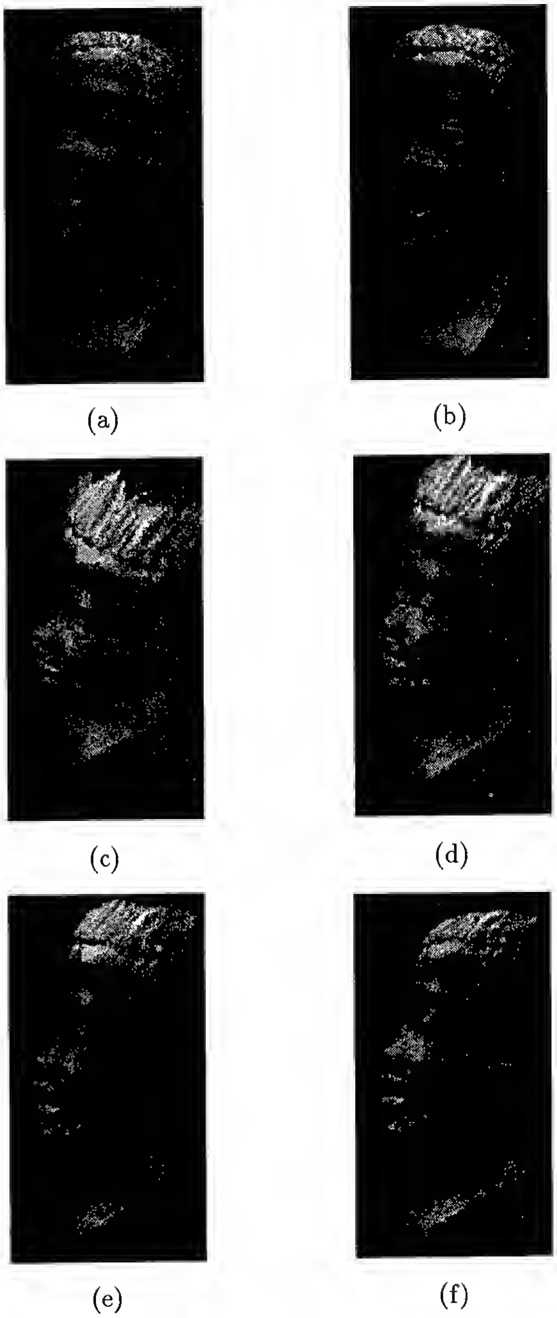
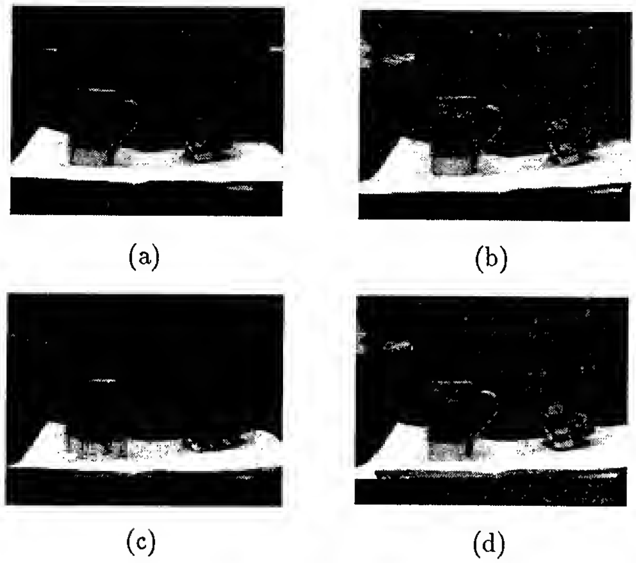


Figure 1: An example of image synthesis using optic-flow and a tensor. The original images are (a) and (b), the rest of the images are synthesized using the method described in the paper.



(a) (b)

(c) (d)

Figure 2: The original two images ((a),(b)). Average of the original two images (c). Average of the two images after rotation cancellation (d).

angles are small. Since both the rank-2 and rank-4 tensors can be decomposed into three homography matrices, the algorithm works the same for two views and three views. Figure 2 shows the two input images as well as an average image of the original images and an average image of the two images after rotation cancellation. For verification we compared our results with the three-view algorithm, by adding a third image (not shown here). The recovered rotation angles differed by less than 0.01 radians. The visual result was indistinguishable.

6 Conclusion

We unified two-view, three-view and, as a result, multi-view geometry with the trilinear tensor as the connecting thread. This was done by developing a basic tensorial operator that describes the change in the tensor elements as a result of camera motion and using it to create a chain of tensors that include the epipole, the fundamental matrix - as a rank-2 trivalent tensor, and the rank-4 trilinear tensor in a single framework. The rank-2 tensor of two views and the rank-4 tensor of three views share the same properties and are governed by a single set of basic tensorial operators. As a result algorithms developed under the three-view paradigm will apply to all camera configurations, be it two or three cameras. Apart from the theoretical result, we showed two practical examples that make use of this theory. An image-based rendering application that uses the basic tensorial operators to seamlessly move from rank-2 tensor (representing the geometry of two views) to rank-4 tensors (representing the geometry of three views), and an image-stabilization algorithm that works unchanged for two or three images.

References

[1] S. Avidan and A. Shashua. Unifying Two-View and Three-View Geometry. Technical Report, TR96-21, Hebrew Univ., December 1996.

[2] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of the European Conference on Computer Vision*, pages 563–578, Santa Margherita Ligure, Italy, June 1992.

[3] O.D. Faugeras. Stratification of three-dimensional vision: projective, affine and metric representations. *Journal of the Optical Society of America*, 12(3):465–484, 1995.

[4] O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995.

[5] R. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the International Conference on Computer Vision*, pages 882–887, Cambridge, MA, June 1995.

[6] Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. In *Proceedings of the European Conference on Computer Vision*, pages 589–599, Stockholm, Sweden, May 1994. Springer Verlag, LNCS 800.

[7] B. Rousso, S. Avidan, A. Shashua and S. Peleg. Robust Recovery of Camera Rotation from Three Frames. In *IEEE Conference on Computer Vision and Pattern Recognition*, San-Francisco, CA., June 1996.

[8] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[9] A. Shashua and S.J. Maybank. Degenerate *n* point configurations of three views: Do critical surfaces exist? Technical report, Hebrew University of Jerusalem, November 1996.

[10] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, 1996.

[11] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *Proceedings of the International Conference on Computer Vision*, June 1995.

Multi-Image Correspondence using Geometric and Structural Constraints

George T. Chou* Seth Teller
MIT Computer Graphics Group
545 Technology Square NE43-208
Cambridge MA 02139
gtc@graphics.lcs.mit.edu, http://www.graphics.lcs.mit.edu

Abstract

In this paper, the problem of recovering three-dimensional information from multiple images is considered. The goal is to build a system that can incrementally process images acquired from arbitrary camera positions. Our approach makes use of both the geometric constraints inherent in the camera configuration, as well as the structural relationships between image features. The correspondence problem is analyzed directly in 3D through multi-image triangulation. To address the possibilities of false features and spurious correspondence, every initial match is modeled as a hypothesis. At the core of our system is a state machine which keeps track of matching hypotheses in various states of certainty, and evolves their states in response to new evidence.

1 Introduction

Traditional multi-image reconstruction systems, both stereo-based and motion-based, assume that successive images are taken from very similar viewpoints and are processed in the order of acquisition. This assumption makes the matching of features in successive images robust, since the appearance of the features changes very little. However, enforcing this assumption makes it difficult to achieve the precision possible by matching features in views of the same scene taken from distant viewpoints at arbitrary times. Moreover, from an engineering point of view, it is very limiting to assume that the thousands of images required for a large-scale reconstruction project would be gathered as a nearly-continuous image stream.

Our goal is to be able to process images in arbitrary order, without placing assumptions on the order of acquisition or strong constraints on the viewpoint. We exploit two other sources of constraints to reduce the feature matching ambiguity introduced by our attempt to achieve this goal. First, we use a combination of sensors, including GPS, to annotate each image with an estimate of its acquiring camera's position and orientation. Therefore, we can estimate, in a coordinate system common to all images, the 3D ray where a feature in one camera view lies. Second, we limit ourselves to urban environments, in which there is an abundance of line and vertex features that are relatively easy to identify from a wide range of viewpoints.

Given these considerations, we have designed a method for the recovery of 3D structure from multiple images of an urban scene. The algorithm operates by establishing long-baseline correspondences between 3D features in distant images. However, just as in 2D, spurious matches can occur in 3D. The occurrence of false matches can be significantly reduced by supplementing geometric constraints of imaging configuration – the camera pose estimates – with knowledge about structural relationship of image features – vertex and edge relationships.

[*] Funding for this research was provided in part by the Advanced Research Projects Agency of the Department of the Defense under Ft. Huachuca contract DABT63-95-C-0009.

Still, feature detection is by no means a perfect process. Each matching hypothesis must be supported by a sufficient number of consistent observations before it can be confirmed. A state machine has been developed to keep track of the hypotheses and their estimated uncertainties.

2 Previous Work

Over the years, stereo researchers have explored countless ways to improve the performance of stereo algorithms. A primary objective is to establish reliable correspondence across two or more images. The challenge is that when a large area must be searched for a match, the potential for spurious matches increases also.

In response to this problem, researchers first turned to coarse-to-fine methods [Grimson, 1981], [Terzopoulos, 1983]. In these systems, matching begins at a low resolution of the image in order to cover large displacements. Matching then proceeds to higher resolutions where results from lower resolutions are used to constrain the search. This class of method cannot deal with significant perspective distortion and occlusion present in long baseline images.

Other researchers advocated using multiple images acquired with closely spaced cameras as a way of extending the baseline of analysis while minimizing false matches [Herman and Kanade, 1986], [Baker and Bolles, 1989]. By exploiting the temporal coherence of very short baseline images, stereo correspondence can be performed accurately through incremental tracking of pixels or features. Although these methods seem to work well, they are dependent on the temporal coherence of the input for reliable feature tracking. They cannot, for example, associate images which are taken at very different times, but which contain observations of identical real-world structures.

Another approach is to utilize the structural relationship between image features to resolve matches [Lim and Binford, 1988], [Horaud and Skordas, 1989]. It has been observed that structural properties tend to be more invariant with respect to viewing changes than local image/feature properties. The problem of correspondence then becomes a problem in finding the mapping which best preserves the structural relationship. Because these methods often assume their feature extraction process as ideal, they tend to be fragile with real images.

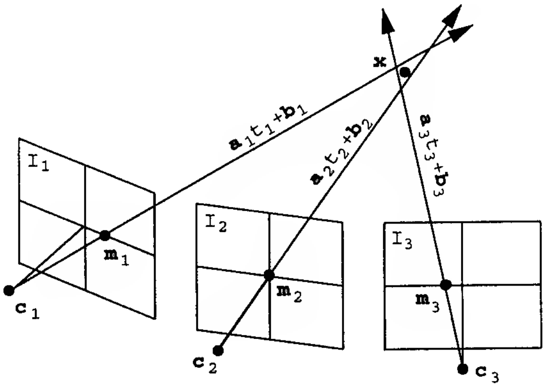


Figure 1: Multi-image triangulation

Recently, new algorithms capable of analyzing long baseline inputs have been proposed. Bedekar and Haralick [1996] describe a method for Bayesian triangulation and hypothesis testing. A major drawback of their work is that they do not consider the possibility of spurious matches. Collins [1996] present a space-sweep approach to multi-image matching. The problem with this method is that it uses a constant threshold for rejecting false matches, and so does not handle underlying causal factors in a generic fashion.

3 Multi-Image Triangulation

The basic principle underlying the recovery of three-dimensional information from two-dimensional images is triangulation. Suppose we are given the corresponding image positions $\mathbf{m}_i$ of a 3D point $\mathbf{x}$ projected onto a set of images I_i. We can compute the 3D position of the point by finding the intersection of rays projected, respectively, from camera $\mathbf{c}_i$ and passing through the image feature $\mathbf{m}_i$ (Figure 1).

Typically the rays will not intersect precisely at one point. However, a well-fitting point $\mathbf{x}$ can be estimated with the least squares method. Our goal is to minimize the sum of squared distances of the rays to point $\mathbf{x}$:

$$D(\mathbf{x}) = \sum_i (\mathbf{a}_i t_i + \mathbf{b}_i - \mathbf{x})^T (\mathbf{a}_i t_i + \mathbf{b}_i - \mathbf{x}) \quad (1)$$

where $\mathbf{a}_i$ is the direction of the ray i, and $\mathbf{b}_i$ is an arbitrary point on the ray (usually taken to be the camera position $\mathbf{c}_i$).

Setting $dD(\mathbf{x})/d\mathbf{x} = 0$, we get

$$\sum_i (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})\, \mathbf{x} \quad (2)$$
$$= \sum_i (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) \mathbf{b}_i$$

We note that this is a linear system, $\mathbf{Ax} = \mathbf{b}$. Using singular value decomposition the matrix $\sum_i (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})$ can be decomposed into

$$\sum_i (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) = \mathbf{UWU}^T \quad (3)$$

where $\mathbf{U}$ is an orthonormal matrix satifying $\mathbf{U}^T = \mathbf{U}^{-1}$, and $\mathbf{W}$ is a diagonal matrix containing the singular values. The least squares estimate $\hat{\mathbf{x}}$ is then

$$\hat{\mathbf{x}} = \mathbf{UW}^{-1}\mathbf{U}^T \left(\sum_i (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I})^T (\mathbf{a}_i \mathbf{a}_i^T - \mathbf{I}) \mathbf{b}_i \right) \quad (4)$$

The residual of the intersection process is given by $D(\hat{\mathbf{x}})$ in Equation (1).

4 Matching via Triangulation

Suppose we hypothesize that a set of image features is in correspondence. A direct method for testing the hypothesis would be to apply multi-image triangulation on the established feature set, and examine the residual of the least square computation. If the residual is greater than a certain threshold, there is no single 3D point near which all of the rays pass, and the correspondence hypothesis can be rejected.

However, we cannot hastily accept any intersection of rays as a match. Figure 2 illustrates a case in point. In the figure, the rays of vertices a_1 and a_2 intersect with the rays of vertices b_1 and b_2 by accident. By themselves, the accidental intersections could be interpreted as a line floating in front of two buildings. This is clearly incorrect, and situations like this are not uncommon. Whenever multiple images are taken with a camera revolving around some region in space, there will be many nearly-crossing rays.

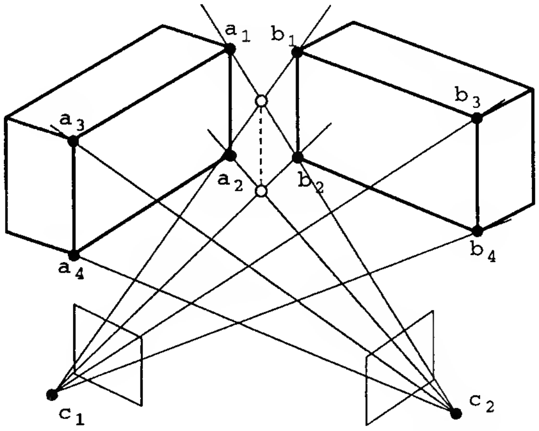


Figure 2: Two spurious matches

Additional observation of the the features is needed for resolving this ambiguity. Certain structural properties, for example adjacency, are invariant with respect to large changes in viewing direction. Connectivity of vertices is a useful structural property in this regard. For two vertices to match, we require that at least two of their incident edges must match also. In the case of Figure 2, we test whether two incident edges of a_1 match with two incident edges of b_1, and can quickly reject this configuration as a spurious intersection. To follow this strategy, we need to determine vertex connectivity information from the images.

Unfortunately, no existing feature extraction algorithm is perfect. We may never be certain that a feature detected from an image is not an artifact of the extraction process. For instance, occlusion often generates incidental features like T-junctions. Since T-junctions are not intrinsic to any real 3D object, their presence can confuse the matching process. To deal with these problems, we treat matches as explicit hypotheses that must later be confirmed by additional matches. We also estimate our level of certainty in each hypothesis and update it as new evidence becomes available.

4.1 The Data Structures

We list here five types of data structures that are relevant to our algorithm. The first two are image features, and the last three are matching hypotheses in increasing states of certainty.

- *2D lines* – are extracted by fitting lines to the output of an edge detector. The system constructs 2D lines only for the purpose of vertex detection.
- *2D vertices* – are located by intersecting 2D lines that form an L-junction. Vertices are the key features used in the correspondence process. Each vertex is described by: 1) a label, 2) an image position, 3) the number of incident lines, and 4) any connected adjacent vertices.
- *2D hypothesis* – is a list of matched 2D vertices with a combined baseline too short to produce a reliable 3D estimate. Each 2D hypothesis is described by: 1) a label, 2) the number of contributing vertices, 3) a list of matched 2D vertices, and 4) baseline information.
- *3D hypothesis* – is a list of matched 2D vertices with a 3D estimate, but a number of observations insufficient for confirmation as a 3D model. Each 3D hypothesis is described in the same way as a 2D hypothesis, with two elements of additional information: 5) a 3D estimate of the feature's position, and 6) an estimate of the reconstruction error in 3D.
- *3D element* – is a confirmed 3D hypothesis. Each 3D model is described in exactly the same way as a 3D hypothesis.

4.2 The Matching Algorithm

The algorithm maintains a set of hypotheses, and evolves the state of each hypothesis after insertion of a new image.

After the features (lines and vertices) of a new image have been extracted, the algorithm tries to find confirming evidence for existing hypotheses among any newly observed 2D features. The algorithm first attempts to reduce reconstruction error for any 3D element for which a new observation is found. Next, hypotheses are processed in order of most to least evolved, beginning with 3D hypotheses, then 2D hypotheses, and finally unmatched 2D vertices. For each, confirmatory evidence is sought among any newly identified features.

For every existing element/hypothesis/vertex:

1. For every new vertex, we project a ray from the new camera position through the new vertex.
2. For 3D elements and 3D hypotheses, we find the shortest distance between the ray and the estimated 3D position of the element/hypothesis. If this distance is sufficiently small, we check to see if at least two incident edges of the element/hypothesis match edges incident to the new vertex.
3. For 2D hypotheses and unmatched vertices, we find the residual resulting from intersecting this ray with rays from all matched vertices in the 2D hypothesis, or the single ray of the unmatched vertex. If this residual is sufficiently small, we check for matching adjacent edges as in Step 2.
4. We link the current model, hypothesis or vertex with the new vertex that gives the best match.

Each new vertex can be matched with more than one hypothesized object. Thus, a spurious match will not affect other objects in the system. After each new match is identified, we test for these possible state transitions (Figure 3):

- 3D hypothesis → 3D element
  if the number of observations is sufficient.
- 2D hypothesis → 3D element
  if the baseline is long enough and the number of observations is sufficient.
- 2D hypothesis → 3D hypothesis
  if the baseline is long enough.
- 2D feature → 3D hypothesis
  if the baseline is long enough.

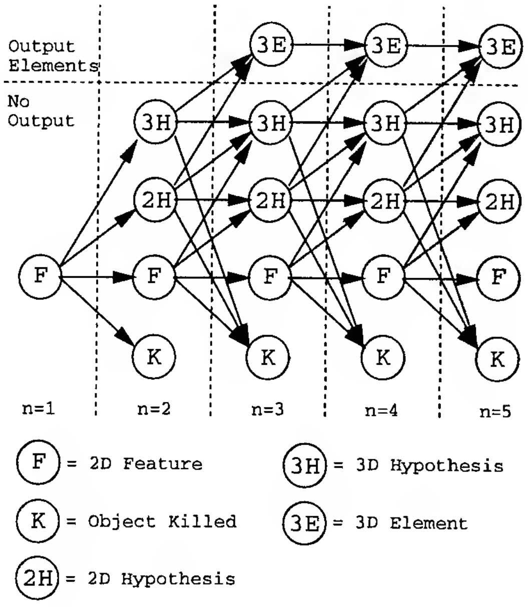


Figure 3: State evolution diagram

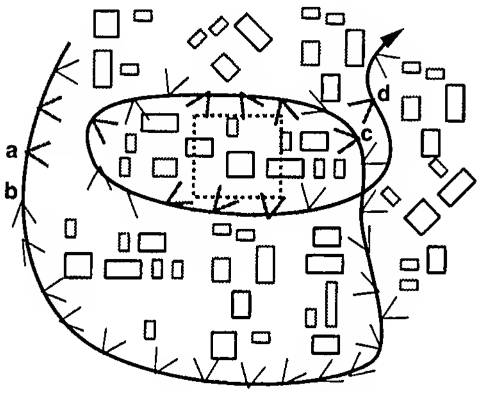


Figure 4: Images are associated spatially, not temporally.

- 2D feature → 2D hypothesis
  if the baseline is not long enough.

If a hypothesis lingers longer than permitted without confirmatory evidence, it is "killed" or deleted from the set of active hypotheses.

5 Image Insertion

Above, we specified the processing to be done for each newly inserted image. Rather than insert the images in temporal order (the order in which they were acquired), we process images in groups according to whether they are suspected to have observed the same region of absolute 3D space (Figure 4). That is, given a set of images annotated with estimates of 6-DOF pose, we fix our attention on a region of 3D space (the dashed box in Figure 4), then identify those images possibly containing observations of this region from a distance less than some absolute threshold (typically, one hundred meters). In the figure, this set of images is represented by bold wedges. These images are inserted in arbitary order and processed as described above, producing a stateful set of feature hypotheses. The region of interest is then moved; any 3D elements no longer in the region of interest are output, and the set of relevant images is coherently updated to contain observations of the new region of interest.

6 Conclusion

In this paper, we describe a method for matching images acquired from arbitrary camera positions. Rather than processing images in temporal order, we process images by grouping them according the 3D regions they observe. The method operates by hypothesizing 3D features, then seeking confirmatory evidence for these features in successively inserted images. This incremental approach seeks to evolve feature hypotheses by amassing a sufficiently large number of observations which agree on a feature's position to within a sufficiently small tolerance or reconstruction error.

References

[Baker and Bolles, 1989] H.H. Baker and R.C. Bolles. Generalizing Epipolar-Plane Image Analysis on the Spatiotemporal Surface. *International Journal of Computer Vision*, 3:33-49, 1989.

[Bedekar and Haralick, 1996] A.S. Bedekar and R.M. Haralick. Finding Corresponding Points Based on Bayesian Triangulation *Proc. IEEE Computer Vision and Pattern Recognition*, San Francisco, CA, 1996, pp61-66.

[Collins, 1996] R.T. Collins. A Space-Sweep Approach to True Multi-Image Matching *Proc. IEEE Computer Vision and Pattern*

Recognition, San Francisco, CA, 1996, pp358-363.

[Grimson, 1981] W.E.L. Grimson. A Computer Implementation of a Theory of Human Stereo Vision. *Phil. Trans. Royal Soc. London*, B292:217-253, 1981.

[Herman and Kanade, 1986] M. Herman and T. Kanade. Incremental Reconstruction of 3D Scenes from Multiple Complex Images. *Artificial Intelligence*, 30(3):289-341.

[Horaud and Skordas, 1989] R. Horaud and T. Skordas. Stereo Correspondence Through Feature Groupings and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168-1180, 1989.

[Lim and Binford, 1988] H. S. Lim and T. O. Binford. Structural Correspondence in Stereo Vision. *Proc. DARPA Image Understanding Workshop.* Cambridge, MA, 1988, pp794-808.

[Terzopoulos, 1983] D. Terzopoulos. Multilevel Computational Processes for Visual Surface Reconstruction. *Computer Vision, Graphics, Image Processing.* 24:52-96, 1983.

A Full–Projective Improvement for Lowe's Pose–estimation Algorithm *

Helder Araújo, Rodrigo L. Carceroni, Chris Brown
University of Coimbra and
Computer Science Department
University of Rochester
Rochester, NY 14627-0226

Abstract

Both the original version of David Lowe's influential and classic algorithm for tracking known objects and a reformulation of it implemented by Ishii *et al.* rely on (different) approximated imaging models. Removing their simplifying assumptions yields a full projective solution with significantly improved accuracy and convergence, and arguably better computation–time properties.

1 Introduction and History

The ability to track a set of points in a moving image plays a fundamental role in several computer vision applications with real time constraints such as autonomous navigation, surveillance, grasping, and manipulation. Often some geometrical invariants of these points (such as their relative spatial positions, in the case of a rigid object) are known in advance. Algebraic solutions with perspective camera models have been proposed for several variations of this problem (references suppressed here). However, the resulting techniques usually work only with a limited number of points and are thus sensitive to additive noise and erroneous matching. Furthermore, they usually depend on numerical techniques for finding zeros of fourth– or fifth–degree polynomial equations.

Pioneering work by Lowe [9; 8; 7] and Gennery [5] addressed the problem in a projective framework. Lowe showed that the direct use of numeric optimization techniques is an effective way to overcome the lack of robustness that makes the traditional analytical techniques infeasible in practice.

DeMenthon and Davis [4; 10] and Christy and Horaud [3] propose techniques that start with weak– or para–perspective solutions, respectively, and refine them iteratively to recover the full perspective pose. Phong, Horaud *et al.* [11] showed that it is possible to decouple completely the recovery of rotational pose parameters from their translational counterparts. However, unlike Lowe's, none of these methods is easily generalizable to deal with uncalibrated focal length or objects (scenes) with internal degrees of freedom.

2 Lowe's Algorithm

Lowe's original algorithm [9; 8; 7] addresses the issue of viewpoint and model parameter computation, given a known 3-D object and the corresponding image. It assumes that the imaging process is a projective transformation. The method can thus be used to identify the pose (translation and orientation with respect to the camera coordinate system) of a local coordinate system affixed to an imaged rigid object. It can also be extended to discover the values of other parameters such as the camera focal length and shape parameters of non–rigid objects. The recovery process is based on the application of Newton's method.

Rather than solving directly for the vector of parameters in the nonlinear system $\mathbf{r}$, Newton's method computes a vector of corrections $\mathbf{x}$ to be subtracted from the current estimate for $\mathbf{r}$ on each iteration. If $\mathbf{r}^{(i)}$ is the parameter vector for iteration i, then:

$$\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \mathbf{x}. \tag{1}$$

Given a vector of error measurements $\mathbf{e}$ between components of the model and the image, we want to solve for a correction vector $\mathbf{x}$ that eliminates this error:

$$\mathbf{J}\,\mathbf{x} = \mathbf{e}, \qquad \text{where: } \mathbf{J}_{ij} = \frac{\partial e_i}{\partial x_j}. \tag{2}$$

The equations used to describe the projection of a three–dimensional model point $\mathbf{p}$ into a two–dimensional image point (u, v) are:

$$(x, y, z)^T = \mathbf{R}\,(\mathbf{p} - \mathrm{t}),$$
$$(u, v) = (\frac{fx}{z}, \frac{fy}{z}), \tag{3}$$

where T denotes transpose, $\mathbf{t}$ is a 3-D translation vector (defined in the model coordinate frame) and $\mathbf{R}$ is a rotation matrix that transforms $\mathbf{p}$ in the original model coordinates into a point (x, y, z) in camera–centered coordinates. These are combined in the second equation with the focal length f to perform perspective projection into an image point (u, v).

The problem is to solve for $\mathbf{t}$, $\mathbf{R}$ and possibly f, given a number of model points and their corresponding locations in an image. In order to apply Newton's method,

*This material is based on work supported by the Luso–American Foundation, Calouste Gulbenkian Foundation, JNICT, CAPES process. BEX 0591/95-5, NSF IIP grant CDA-94-01142 and DARPA contract MDA972-92-J-1012.

we must be able to calculate the partial derivatives of u and v with respect to each of the unknown parameters. Lowe [8] proposes a reparameterization of the projection equations, to simplify the calculation by "express[ing] the translations in terms of the camera coordinate system rather than model coordinates":

$$(x', y', z')^T = \mathbf{R}\,\mathbf{p},$$
$$(u, v) = (\frac{fx'}{z'+D_z}+D_x,\ \frac{fy'}{z'+D_z}+D_y). \tag{4}$$

The variables $\mathbf{R}$ and f remain the same as in the previous transform, but vector $\mathbf{t}$ has been replaced by the parameters D_x, D_y and D_z. The two transforms are equivalent when:

$$\mathbf{t} = \mathbf{R}^{-1}\left[-\frac{D_x(z'+D_z)}{f},\ -\frac{D_y(z'+D_z)}{f},\ -D_z\right]^T. \tag{5}$$

According to Lowe, "in the new parameterization, D_x and D_y simply specify the location of the object on the image plane and D_z specifies the distance of the object from the camera". To compute the partial derivatives of the error with respect to the rotation angles (ϕ_x, ϕ_y and ϕ_z are the rotation angles about x, y and z, respectively), it is necessary to calculate the partial derivatives of x, y and z with respect to these angles. Table 1 gives these derivatives for all combinations of variables.

	x	y	z
ϕ_x	0	$-z'$	y'
ϕ_y	z'	0	$-x'$
ϕ_z	$-y'$	x'	0

Table 1: The partial derivatives of x, y and z with respect to counterclockwise rotations ϕ (in radians) about the coordinate axes.

Newton's method is carried out by calculating the optimum correction rotations $\Delta\phi_x$, $\Delta\phi_y$ and $\Delta\phi_z$ to be made about the camera–centered axes. Given Lowe's parameterization, the partial derivatives of u and v with respect to each of the seven parameters of the imaging model (including the focal length f) are given by Table 2.

	u	v
D_x	1	0
D_y	0	1
D_z	$-fc^2x'$	$-fc^2y'$
ϕ_x	$-fc^2x'y'$	$-fc(z'+cy'^2)$
ϕ_y	$fc(z'+cx'^2)$	$fc^2x'y'$
ϕ_z	$-fcy'$	fcx'
f	cx'	cy'

Table 2: The partial derivatives of u and v with respect to each of the camera viewpoint parameters and the focal length, according to Lowe's original approximation. Here $c = \frac{1}{z'+D_z}$.

Lowe then notes that each iteration of the multi–dimensional Newton's method solves for a vector of corrections

$$\mathbf{x} = [\Delta D_x, \Delta D_y, \Delta D_z, \Delta\phi_x, \Delta\phi_y, \Delta\phi_z]^T. \tag{6}$$

Lowe's algorithm dictates that for each point in the model matched against some corresponding point in the image, we first project the model point into the image using the current parameter estimates and then measure the error in the resulting position with respect to the given image point. The u and v components of the error can be used independently to create separate linearized constraints. Making use of the u component of the error, E_u, we create an equation that expresses this error as the sum of the products of its partial derivatives times the unknown error–correcting values:

$$\frac{\partial u}{\partial D_x}\Delta D_x + \frac{\partial u}{\partial D_y}\Delta D_y + \frac{\partial u}{\partial D_z}\Delta D_z +$$
$$\frac{\partial u}{\partial \phi_x}\Delta\phi_x + \frac{\partial u}{\partial \phi_y}\Delta\phi_y + \frac{\partial u}{\partial \phi_z}\Delta\phi_z = E_u. \tag{7}$$

The same point yields a similar equation for its v component. Thus each point correspondence yields two equations. As Lowe says: "from three point correspondences we can derive six equations and produce a complete linear system which can be solved for all six camera–model corrections".

3 Lowe's Approximation

Lowe's formulation assumes that D_x and D_y are constants to be determined by the iterative procedure, when in fact they are not constants at all — they depend on the location of the points being imaged.

Let the translation vector, the rotation matrix and the description of an arbitrary feature in the object frame be denoted, respectively, by:

$$\begin{aligned} \mathbf{t} &= [t_x, t_y, t_z]^T, \\ \mathbf{R} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \\ \mathbf{p} &= [p_1, p_2, p_3]^T. \end{aligned}$$

Then using the projective transformation formulated in Eq. (3) the new parameters D_x, D_y, D_z are given by:

$$D_z = -(r_{31}t_x + r_{32}t_y + r_{33}t_z), \text{ and then:}$$

$$\begin{aligned} D_x &= -f\frac{r_{11}t_x + r_{12}t_y + r_{13}t_z}{(r_{31}p_1 + r_{32}p_2 + r_{33}p_3) + D_z}, \\ D_y &= -f\frac{r_{21}t_x + r_{22}t_y + r_{23}t_z}{(r_{31}p_1 + r_{32}p_2 + r_{33}p_3) + D_z}. \end{aligned} \tag{8}$$

D_z is dependent only on the object pose parameters, but D_x and D_y are also a function of each point's coordinates in the object coordinate frame. It is therefore in general impossible to find a single consistent value either for D_x or for D_y. In the general case both these parameters will depend on the position of each individual object feature. They are not constants — they are only the same for those points for which $r_{31}p_1 + r_{32}p_2 + r_{33}p_3$ has the same value. Therefore we can not use D_x and D_y

as defined in Eq. (4). The assumption that is implicit in Lowe's algorithm as published is that the corrections needed for the translation are much larger than those due to rotation of the object. However, if no restrictions are imposed, the coordinates of the points in the object coordinate frame **p** can assume high values. Even if they do not, the term $r_{31}p_1 + r_{32}p_2 + r_{33}p_3$ may change significantly (due to the object's own geometry) and affect the estimation process.

Ishii's formulation [6] contains different simplifications. See [2] for details.

4 Full Projective Solution

Initially define x', y' and z' as in Lowe's formulation:

$$(x', y', z')^T = \mathbf{R}\,\mathbf{p}.$$

Model the image formation process by Eq. (3). Remove the approximations of Lowe and Ishii by defining:

$$\begin{aligned} D_x^{'} &= -(r_{11}t_x + r_{12}t_y + r_{13}t_z), \\ D_y^{'} &= -(r_{21}t_x + r_{22}t_y + r_{23}t_z), \qquad (9) \\ D_z^{'} &= -(r_{31}t_x + r_{32}t_y + r_{33}t_z). \end{aligned}$$

In this case the image coordinates of each point are given by:

$$(u, v) = \left(f\frac{x'+D_x^{'}}{z'+D_z^{'}},\ f\frac{y'+D_y^{'}}{z'+D_z^{'}} \right). \qquad (10)$$

The partial derivatives of u and v with respect to each of the six pose parameters and the focal length are given by Table 3.

	u	v
$D_x^{'}$	fc	0
$D_y^{'}$	0	fc
$D_z^{'}$	$-fac^2$	$-fbc^2$
ϕ_x	$-fac^2y'$	$-fc(z'+bcy')$
ϕ_y	$fc(z'+acx')$	fbc^2x'
ϕ_z	$-fcy'$	fcx'
f	ac	bc

Table 3: The partial derivatives of u and v with respect to each of the camera viewpoint parameters and the focal length according to our full projective solution. Here $(a, b, c) = (x'+D_x^{'},\ y'+D_y^{'},\ \frac{1}{z'+D_z^{'}})$.

As in Lowe's formulation, the translation vector is computed using Eq. (5), with $D_x^{'}$, $D_y^{'}$ and $D_z^{'}$ as defined in Eq. (9). This translation vector is defined in the object coordinate frame. The minimization process yields estimates of $D_x^{'}$, $D_y^{'}$ and $D_z^{'}$, which are the result of the product of the rotation matrix by the translation vector.

A numerically equivalent but conceptually more elegant way of looking at this solution is through a redefinition of the image formation process, so that rotation and translation are explicitly decoupled, and the translation vector is defined in the camera coordinate frame. Redefine:

$$(x, y, z)^T = \mathbf{R}\,\mathbf{p} + \mathbf{t}, \qquad (11)$$

then: $(D_x^{'}, D_y^{'}, D_z^{'})^T = \mathbf{t}$,

and Eqs. (9) and (10) can be collapsed into:

$$(u, v) = \left(f\frac{x'+t_x}{z'+t_z},\ f\frac{y'+t_y}{z'+t_z} \right). \qquad (12)$$

In this case, the least–squares minimization procedure gives the estimates of the translation vector directly.

5 Experimental Results

We compare the three algorithms described in the previous sections with extensive experiments on synthetic data[1]. This paper is an abbreviated version of [2] and [1], which can be consulted for more details and many more results.

Our experiments take the imaged object to be the eight corners of a cube, with edge lengths equal to 25 times the focal length of the camera (for a 20 mm lens, for instance, this corresponds to a half meter wide, long and deep object). The parameters explicitly controlled, in general, were the depth of the object's center with respect to the camera frame (z_{true}), measured in focal lengths, and the magnitudes of the translation (t_{diff}) and the rotation (r_{diff}) needed to align the initial solution with the true pose. We tested near, intermediate and far viewing situations. The other nine pose and initial solution parameters are in general sampled uniformly over their whole domain.

For each test we compute two global image–space error measures, assuming known correspondence between image and model features. The first, called *Norm of Distances Error* (**NDE**), is the norm of the vector of distances between the positions of the features in the actual image and the positions of the same features in the reprojected image generated by the estimated pose. The second, called *Maximum Distance Error* (**MDE**), is the greatest absolute value of the vector of error distances. Both measures are always expressed using the focal length as length unit.

NDE and MDE do not necessarily indicate how close the estimated pose is from the true pose. We also record individual errors for six different pose parameters: the errors in the x, y and z coordinates of the estimate for the actual object translation vector, measured as relative errors with respect to the object's center actual depth (z_{true}), and the absolute errors in the estimates for the roll, pitch and yaw angles of the object frame with respect to the camera, measured in units of π radians.

For each of the eight different error measures, we compute the average, standard deviation, and median. Statistics that leave out the tails of the error distributions are included to be fair to a method (if any) that underperforms in a few exceptional situations but is better "in general". For more error measures and more statistics see [2; 1].

5.1 Convergence in the General Case

Here we compare the speed of convergence and final accuracy of each method with arbitrary poses and initial

[1] Matlab code implementing the three algorithms is available from the authors

conditions. The statistics for the NDE, based on 13,500 executions per method, plotted in Fig. 1. They show that for most poses Lowe's original approximation converges to a very high global error level and Ishii's approximation only improves the initial solutions in its first iteration and diverges after that. The full projective solution, on the other hand, converges at a superexponential rate to an error level roughly equivalent to the relative rounding error of double precision, which is about 1.11×10^{-16}.

Even taking into account the worst data, our approximation still converges superexponentially to this maximum precision level — the bad cases only slow convergence a bit. In this case Lowe's original algorithm and (especially) Ishii's approximation tend to diverge, yielding some solutions worse than the initial conditions.

The statistics for the errors in the individual pose parameters make the superiority of the full projective approach even more clear. Fig. 2 exhibits the relative errors in the value of the x translation. Both Lowe's and Ishii's algorithms diverge in most situations, while the full projective solution keeps its superexponential convergence. Due to their simplifications, Lowe's and Ishii's methods in those cases are not able to recover the true rotation of the object. They tend to make corrections in the translation components to fit the erroneously rotated models to the image in a least–squares sense, generating very imprecise values for the parameters themselves. Ishii's approximation tends to translate the object as far away from the camera as possible, so that the reprojected images for all points are collapsed into a single spot that minimizes the mean of the squared distances with respect to the true images. Similar results were obtained for the other five parameter–space errors.

5.2 Other Conditions

We ran many tests only reported in brief prose here (see [2] for details). First we assume that the pose is approximately known: In this case the accuracy of Ishii's approximation is much improved (predictably, given its semantics). Instead of diverging, now it converges exponentially towards the rounding error lower bound. It is still dominated by the full projective solution, which converges super–exponentially (in about 5 iterations) for the NDE (and also for all other error metrics tested.)

Timing tests with optimized Matlab code show that the full projective solution (which has only four floating point operations more than Lowe's method in the inner loop) takes 2.99% to 4.21% longer than Lowe's method. However, the standard deviations of the execution times for Lowe's solution are between 6% and 130% bigger than those of the full projective. Thus, the full projective approach may be more suitable for hard real time constraints, due to its smaller sensitivity to ill–conditioned configurations.

We tested the sensitivity of the techniques to individual variations in the depth of the object along the optical axis and in the magnitudes of the translational and rotational errors in the initial solution. We tested the effects of gaussian noise with zero mean and controlled standard deviation added to the coordinates of the image features. The microstructure of all these results is interesting and explicable [2] but the message is the same: the full projective solution performs significantly, qualitatively better for all cases when any solution is usable.

5.3 Accuracy in Practice

Would the projective method perform better in a practical situation? The noise experiments were relevant, but in a real system a more important issue is establishing the initial conditions. One could use a smoothing filter, but this approach is very dependent on application–specific parameters, such as the sampling rate of the camera, the bandwidth of the image processing system as a whole, the positional depth, the linear speed and the angular speed of the tracked object.

We follow a more general approach: use a weaker camera model to generate an initial solution for the problem analytically, and then use the projective iterative solution(s) to refine this initial estimate. This approach was suggested by DeMenthon and Davis [4], who introduced a way of describing the discrepancy between a weak perspective solution and the full perspective pose with a set of parameters that can then be refined numerically, yielding the latter from the former. Let $\mathbf{p}_i$ be the description of the i–th model point in the model frame and $[u_i, v_i]$ be the corresponding image, $1 \leq i < n$. Then, the weak perspective solution proposed in that paper amounts to solving the following set of equations (in a least–squares sense), for the unknown three-dimensional vectors $\mathbf{I}$ and $\mathbf{J}$:

$$\begin{aligned} (\mathbf{p}_i - \mathbf{p}_0) \cdot \mathbf{I} &= u_i - u_0, \quad 1 \leq i < n \\ (\mathbf{p}_i - \mathbf{p}_0) \cdot \mathbf{J} &= v_i - v_0, \quad 1 \leq i < n \end{aligned} \tag{13}$$

A normalization of these vectors yields the two first rows of the rotation component of the transformation that describes the object frame in the camera coordinate system. The third row can then be obtained with a single cross product operation. After that, the recovery of the translation is straightforward.

However, this simple weak perspective approximation introduces errors that increase proportionally not only with the inverse depth of the object, but also with its "off–axis" angle. In order to avoid this problem, we first preprocessed the image to simulate a rotation that puts the center of the object's image in the intersection of the optical axis with the image plane (this seems to be a novel wrinkle in this context). Let the center of the object image be described by $[u, v]$. Then, this transformation, as suggested in [12], is given by:

$$R = \begin{bmatrix} \frac{1}{d_1} & 0 & -\frac{u}{d_1} \\ \frac{u\,v}{d_1 d_2} & \frac{d_1}{d_2} & -\frac{v}{d_1 d_2} \\ \frac{u}{d_2} & \frac{v}{d_2} & \frac{1}{d_2} \end{bmatrix}, \text{where:} \tag{14}$$

$$d_1 = \sqrt{u^2 + 1}, \quad d_2 = \sqrt{u^2 + v^2 + 1}.$$

After this preprocessing, we applied the technique described by Eq. (13), in order to recover the "foveated" pose. Then, we premultiplied the resulting transformation by the inverse of the matrix defined in Eq. (14), in order to recover the original weak perspective pose,

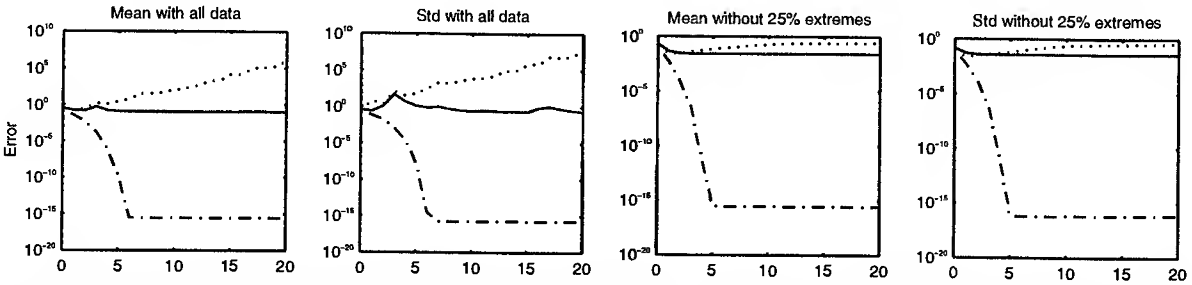


Figure 1: Convergence of an image–space error metric, the Norm of Distances Error (see introduction of Section 5), with respect to the number of iterations of Lowe's (solid line), Ishii's (dotted line), and our full projective solution (dash–dotted line). Tests performed with a cube, rotated by arbitrary angles with respect to the camera frame.

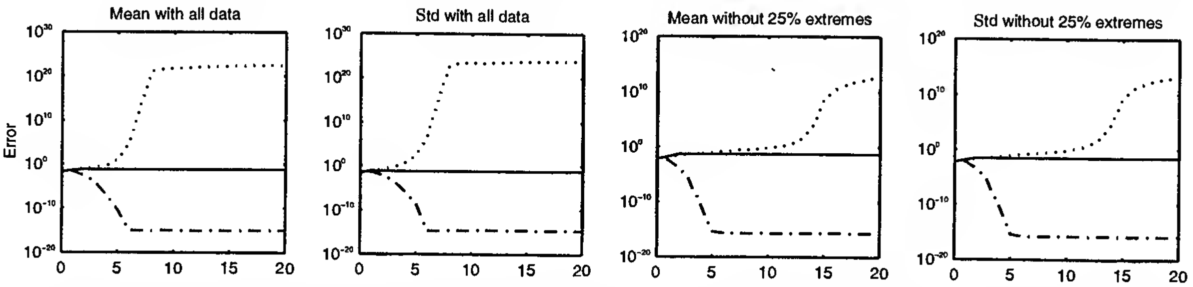


Figure 2: Convergence of the ratio between the error on the estimated x translation and the actual depth of the object's center, with respect to the number of iterations of Lowe's (solid line), Ishii's (dotted line), and our full projective solution (dash–dotted line). Tests performed with a cube, rotated by arbitrary angles with respect to the camera frame.

which was used as the initial solution for the iterative techniques in being compared.

The only controlled parameter left was the actual depth of the object's center (z_{true}). We chose nine average values for it, growing exponentially from 25 to 6,400 focal lengths. The noise standard deviation was set at 0.002 focal lengths (corresponding roughly to a 512 × 512 spatial quantization). The number of iterations of each method per run was set at 2, allowing a real time execution rate of about 100 Hz. For each average value of z_{true}, 2,500 independent runs of each technique were performed.

The statistics for the NDE, depicted in Fig. 3, show that our full projective solution was up to one order of magnitude more accurate than the other two methods for most cases in which the distance was smaller than 1,000 focal lengths (about 20 m, with the typical focal length of 20 mm). For distances bigger than that, the precision of the weak perspective initial solution alone was bigger than the limitation imposed by the noise and so the three techniques performed equally well.

For x translation error (Fig. 4) and the other five parameter–space errors, we found that all the techniques exhibit parameter–space accuracy peaks in the range of 50 to 400 focal lengths. When the object is too close, the quality of the initial weak perspective solution degrades quickly. When the object is too far away, the noise gradually overpowers the information about both the distance (via observed size) and the orientation of the object, since all the feature images tend to collapse into a single point.

Similar results were obtained when the number of iterations for each run was raised to 5. This suggests that our solution may be very well suited for indoor applications in which it is possible to keep a safe distance between the objects of interest and the camera.

6 Discussion and Conclusion

This note formulates a full projective treatment of a pose– or parameter–recovery algorithm initially proposed by Lowe [7; 8; 9]. The projective formulation is compared with formulations by Lowe and Ishii [6] that approximate the full projective case. Many experiments based on different scenaria are presented here, and more are available in [2]. Our experiments indicate that a straightforward reformulation of the perspective imaging equations removes mathematical approximations that limit the precision of Lowe's and Ishii's formulations. The full projective algorithm has better accuracy with a minimal increase in terms of computational cost per iteration.

The full projective solution is very stable for a wide range of actual object poses and initial conditions. In some particularly extreme scenaria, our approach does suffer from numerical stability problems, but in these situations the accuracy of Lowe's and Ishii's approximations is also unacceptable, with errors of one or more orders of magnitude in the values of the pose parameters. We believe that this type of problem is a consequence of Newton's method and can only be overcome with the use of more powerful numerical optimization techniques, such as trust region methods.

In scenaria that may realistically arise in applications such as indoor navigation, with the use of reasonable

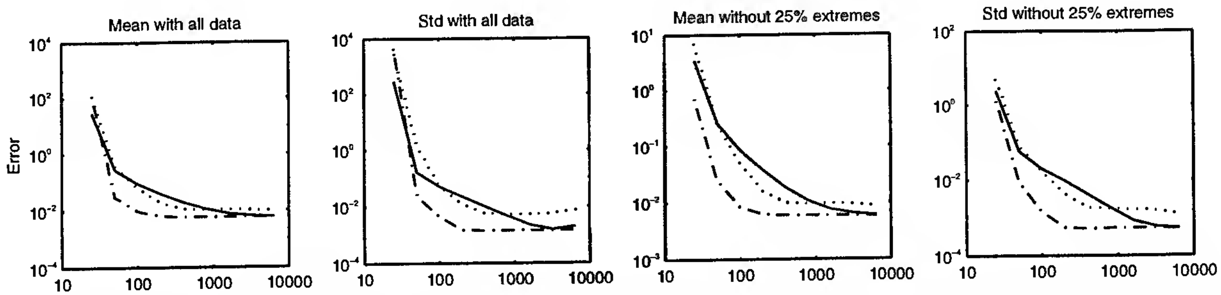


Figure 3: Sensitivity of an image–space error metric, the Norm of Distances Error (see introduction of Section 5), with respect to the actual depth of the object's center (in focal lengths), for Lowe's (solid line), Ishii's (dotted line), and our full projective solution (dash–dotted line). Tests performed with initial solutions generated by a weak perspective approximation.

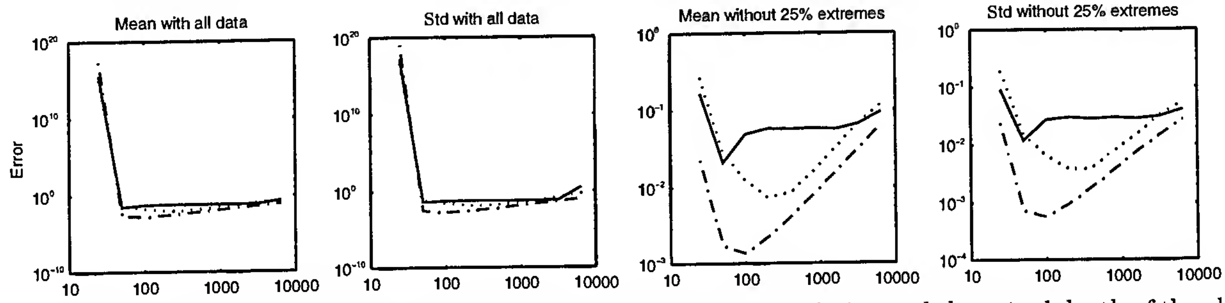


Figure 4: Sensitivity of the ratio between the error on the estimated x translation and the actual depth of the object's center, with respect to the actual depth of the object's center (in focal lengths), for Lowe's (solid line), Ishii's (dotted line), and our full projective solution (dash–dotted line). Tests performed with initial solutions generated by a weak perspective approximation.

(weak perspective) initial solutions and taking into account the effect of additive gaussian noise in the imaging process, the full projective formulation outperforms both Lowe's and Ishii's approximations by up to an order of magnitude in terms of accuracy, with practically the same computational cost.

References

[1] H. Araujo, R. L. Carceroni, and C. M. Brown. A full projective formulation to improve the accuracy of lowe's pose–estimation algorithm.

[2] H. Araujo, R. L. Carceroni, and C. M. Brown. A fully projective formulation for lowe's tracking algorithm. Technical Report 641, University of Rochester Computer Science Dept., Nov 1996.

[3] S. Christy and R. Horaud. Euclidean reconstruction: from paraperspective to perspective. In *Proc. 4th European Conference on Computer Vision*, volume 2, pages 129–140, 1996.

[4] D. F. DeMenthon and L. S. Davis. Model–based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

[5] Donald B. Gennery. Visual tracking of known three–dimensional objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[6] M. Ishii, S. Sakane, M. Kakikura, and Y. Mikami. A 3-d sensor system for teaching robot paths and environments. *International Journal of Robotics Research*, 6(2):45–59, 1987.

[7] David Lowe. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pages 121–127, College Park, MD, Apr 1980.

[8] David Lowe. Three–dimensional object recognition from single two–dimensional images. *Artificial Intelligence*, 31(3):355–395, Mar 1987.

[9] David Lowe. Fitting parameterized three–dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[10] D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495–511, May 1996.

[11] T. Q. Phong, R. Horaud, and P. D. Tao. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, 15:225–243, 1995.

[12] Y. Wu, S. S. Iyengar, R. Jain, and S. Bose. A new generalized computational framework for finding object orientation using perspective trihedral angle constraint. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(10):961–975, Oct 1994.

View Morphing: Uniquely Predicting Scene Appearance from Basis Images

Steven M. Seitz Charles R. Dyer*
Department of Computer Science
University of Wisconsin
Madison, WI 53706
E-MAIL {seitz|dyer}@cs.wisc.edu
HOMEPAGE http://www.cs.wisc.edu/~{seitz|dyer}

Abstract

This paper analyzes the conditions when a discrete set of images implicitly describes scene appearance for a continuous range of viewpoints. It is shown that two basis views of a static scene uniquely determine the set of all views on the line between their optical centers when a visibility constraint is satisfied. Additional basis views extend the range of predictable views to 2D or 3D regions of viewpoints. A simple scanline algorithm called *view morphing* is presented for generating these views from a set of basis images. The technique is applicable to both calibrated and uncalibrated images.

1 Introduction

Image-based representations of 3D scenes are currently being developed by many researchers in the computer vision and computer graphics communities. These representations encode scene appearance with a set of images that are adaptively combined to produce new views of the scene. At the heart of this area lies a fundamental question: To what extent can scene appearance be modeled with a sparse set of images? Clearly, the images provide scene appearance at a discrete set of viewpoints but it has not been clear if a more complete coverage of viewspace is theoretically possible. A number of "view synthesis" techniques have been developed recently [Chen and Williams, 1993, Laveau and Faugeras, 1994, McMillan and Bishop, 1995, Beymer and Poggio, 1996] to extend the range of predictable views. However, those methods require solving ill-posed correspondence tasks, suggesting that the view synthesis problem is inherently ill-posed.

As a foundation for work in this area we feel it is necessary to answer the following two questions. First, given two perspective views of a static scene, under what conditions can new views be predicted? Second, *which* views are determined from a set of basis images? In this paper we show that a specific range of perspective views is theoretically determined from two or more basis views, under a generic visibility assumption called *monotonicity*. This result applies when either the relative camera configurations are known or when only the fundamental matrix is available. In addition, we present a simple technique for generating this particular range of views using image interpolation. Importantly, the method relies only on *measurable* image information, avoiding ill-posed correspondence problems entirely. Furthermore, all processing occurs at the scanline level, effectively reducing the original 3D synthesis problem to a set of simple 1D image transformations that can be implemented efficiently on existing graphics workstations. The work presented here extends to perspective projection previous results on the orthographic case [Seitz and Dyer, 1995].

*The support of the Defense Advanced Research Projects Agency, and the National Science Foundation under Grant No. IRI-9530985 is gratefully acknowledged.

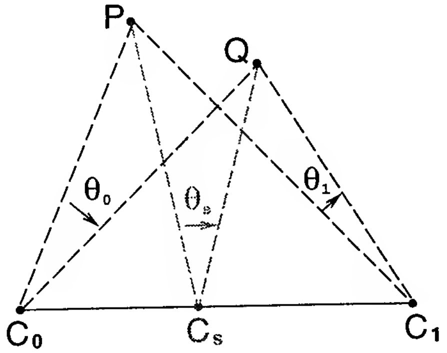


Figure 1: The monotonicity constraint holds when $\theta_0\theta_1 > 0$ for all pairs of scene points $\mathbf{P}$ and $\mathbf{Q}$ in the same epipolar plane.

We begin by introducing the *monotonicity* constraint and describing its implications for view synthesis in Section 2. Section 3 considers *how* views can be synthesized, and describes a simple and efficient algorithm called *view morphing* for synthesizing new views by interpolating images, under the assumption that the relative geometry of the two cameras is known. Section 4 investigates the case where the images are *uncalibrated*, i.e., the camera geometry is unknown. Section 5 presents extensions when three or more basis views are available. Section 6 presents some results on real images.

2 View Synthesis and Monotonicity

Can the appearance from new viewpoints of a static three-dimensional scene be predicted from a set of basis views of the same scene? One way of addressing this question is to consider view synthesis as a two-step process—reconstruct the scene from the basis views using stereo or structure-from-motion methods and then reproject to form the new view. The problem with this paradigm is that view synthesis becomes at least as difficult as 3D scene reconstruction. This conclusion is especially unfortunate in light of the fact that 3D reconstruction from sparse images is generally ambiguous—a number of different scenes may be consistent with a given set of images; it is an ill-posed problem. This suggests that view synthesis is also ill-posed.

In this section we present an alternate paradigm for view synthesis that avoids 3D reconstruction and dense correspondence as intermediate steps, instead relying only on *measurable* quantities, computable from a set of basis images. We first consider the conditions under which reconstruction is ill-posed and then describe why these conditions do not impede view synthesis. Ambiguity arises within regions of uniform intensity in the images. Uniform image regions provide shape and correspondence information only at boundaries. Consequently, 3D reconstruction of these regions is not possible without additional assumptions. Note however that boundary information is sufficient to predict the appearance of these regions in new views, since the region's interior is assumed to be uniform. This argument hinges on the notion that uniform regions are "preserved" in different views, a constraint formalized by the condition of *monotonicity* which we introduce next.

Consider two views, V_0 and V_1, with respective optical centers $\mathbf{C}_0$ and $\mathbf{C}_1$, and images I_0 and I_1. Denote $\overline{\mathbf{C}_0\mathbf{C}_1}$ as the line segment connecting the two optical centers. Any point $\mathbf{P}$ in the scene determines an epipolar plane containing $\mathbf{P}$, $\mathbf{C}_0$, and $\mathbf{C}_1$ that intersects the two images in conjugate epipolar lines. The monotonicity constraint dictates that all visible scene points appear in the same order along conjugate epipolar lines of I_0 and I_1. This constraint is used commonly in stereo matching because the fixed relative ordering of points along epipolar lines simplifies the correspondence problem. Despite its usual definition with respect to epipolar lines and images, monotonicity constrains only the location of the optical centers with respect to points in the scene—the image planes may be chosen arbitrarily. An alternate definition that isolates this dependence more clearly is shown in Figure 1. Any two scene points $\mathbf{P}$ and $\mathbf{Q}$ in the same epipolar plane determine angles θ_0 and θ_1 with the optical centers $\mathbf{C}_0$ and $\mathbf{C}_1$. The monotonicity constraint dictates that for all such points θ_0 and θ_1 must be nonzero and of equal sign. The fact that no constraint is made on the image planes is of primary importance for view synthesis because it means that *monotonicity is preserved under homographies*,

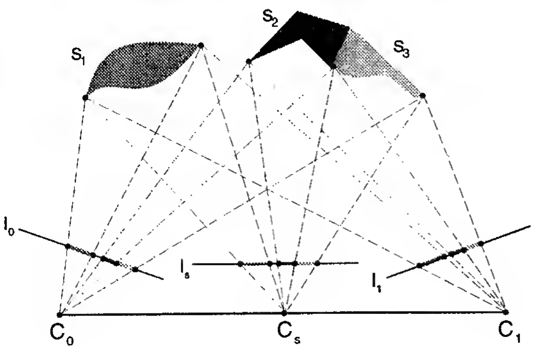


Figure 2: Although the projected intervals in l_0 and l_1 do not provide enough information to reconstruct S_1, S_2 and S_3, they are sufficient to predict the appearance of l_s.

i.e., under image reprojection. This fact will be essential in the next section for developing an algorithm for view synthesis.

A useful consequence of monotonicity is that it extends to cover a continuous range of views in-between V_0 and V_1. We say that a third view V_s is *in-between* V_0 and V_1 if its optical center $\mathbf{C}_s$ is on $\overline{\mathbf{C}_0\mathbf{C}_1}$. Observe that monotonicity is violated only when there exist two scene points, $\mathbf{P}$ and $\mathbf{Q}$, in the same epipolar plane such that the infinite line $\mathbf{PQ}$ through $\mathbf{P}$ and $\mathbf{Q}$ intersects $\overline{\mathbf{C}_0\mathbf{C}_1}$. But $\mathbf{PQ}$ intersects $\overline{\mathbf{C}_0\mathbf{C}_1}$ if and only if it intersects either $\overline{\mathbf{C}_0\mathbf{C}_s}$ or $\overline{\mathbf{C}_s\mathbf{C}_1}$. Therefore monotonicity applies to in-between views as well, i.e., signs of angles are preserved and visible scene points appear in the same order along conjugate epipolar lines of all views along $\overline{\mathbf{C}_0\mathbf{C}_1}$. We therefore refer to the range of views with centers on $\overline{\mathbf{C}_0\mathbf{C}_1}$ as a *monotonic range* of viewspace. Notice that this range gives a lower bound on the range of views for which monotonicity is satisfied in the sense that the latter set contains the former. For instance, in Figure 1 monotonicity is satisfied for all views on the open ray from the point $\mathbf{C}_0\mathbf{C}_1 \bigcap \mathbf{PQ}$ through both camera centers. However, without *a priori* knowledge of the geometry of the scene, we can infer only that monotonicity is satisfied for the range $\overline{\mathbf{C}_0\mathbf{C}_1}$.

The property that monotonicity applies to in-between views is quite powerful and is sufficient to completely predict the appearance of the visible scene from all viewpoints along $\overline{\mathbf{C}_0\mathbf{C}_1}$. Consider the projections of a set of uniform Lambertian surfaces (each surface has uniform radiance, but any two surfaces can have different radiances) into views V_0 and V_1. Figure 2 shows cross sections S_1, S_2, and S_3 of three such surfaces projecting into conjugate epipolar lines l_0 and l_1. Each connected cross section projects to a uniform interval (i.e., an interval of uniform intensity) of l_0 and l_1. The monotonicity constraint induces a correspondence between the endpoints of the intervals in l_0 and l_1, determined by their relative ordering. The points on S_1, S_2, and S_3 projecting to the interval endpoints are determined from this correspondence by triangulation. We will refer to these scene points as *visible endpoints* of S_1, S_2, and S_3.

Now consider an in-between view, V_s, with image I_s and corresponding epipolar line l_s. As a consequence of monotonicity, S_1, S_2, and S_3 project to three uniform intervals along l_s, delimited by the projections of their visible endpoints. Notice that the intermediate image does not depend on the specific shapes of surfaces in the scene, only on the positions of their visible endpoints. **Any number of distinct scenes could have produced I_0 and I_1, but each one would also produce the same set of intermediate images.** Hence, all views along $\overline{\mathbf{C}_0\mathbf{C}_1}$ are determined from I_0 and I_1. This result demonstrates that view synthesis under monotonicity is an inherently well-posed problem—and is therefore much easier than 3D reconstruction and related motion analysis tasks requiring smoothness conditions and regularization techniques.

A final question concerns the *measurability* of monotonicity. That is, can we determine if two images satisfy monotonicity by inspecting the images themselves or must we know the answer *a priori*? Strictly speaking, monotonicity is not measurable in the sense that two images may be consistent with multiple scenes, some of which satisfy monotonicity and others that do not. However, we can determine whether or not two images are *consistent* with a scene for which monotonicity applies, by checking that each epipolar line in the first image is a monotonic warp of its conjugate in the second image.

3 View Morphing

The previous section established that certain views are determined from two basis views under an assumption of monotonicity. In this section we present a simple approach for synthesizing these views based on image interpolation. The procedure takes as input two images, I_0 and I_1, their respective projection matrices, $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$, and a third projection matrix $\mathbf{\Pi}_s$ representing the configuration of a third view along $\overline{\mathbf{C}_0\mathbf{C}_1}$. The result is a new image I_s representing how the visible scene appears from the third viewpoint.

We begin with a special case where the image planes are parallel and aligned with $\overline{\mathbf{C}_0\mathbf{C}_1}$. This configuration is often used in stereo applications and will be referred to as the *parallel configuration.* The situation is expressed algebraically using the projection equations as follows. A camera is represented by a 3×4 homogeneous matrix $\mathbf{\Pi} = [\mathbf{H} \mid -\mathbf{HC}]$. The optical center is given by $\mathbf{C}$ and the image plane normal is the last row of $\mathbf{H}$. A scene point (X, Y, Z) is expressed in homogeneous coordinates as $\mathbf{P} = [X\ Y\ Z\ 1]^T$ and an image point (x, y) by $\mathbf{p} = [x\ y\ 1]^T$. Because homogeneous structures are invariant under scalar multiplication, $s\mathbf{P}$ and $\mathbf{P}$ represent the same point, and similarly for $s\mathbf{p}$ and $\mathbf{p}$. We therefore reserve the notation $\mathbf{P}$ and $\mathbf{p}$ for points whose last coordinate is 1. All other multiples of these points will be denoted as $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{p}}$. The perspective projection equation is:

$$\tilde{\mathbf{p}} = \mathbf{\Pi P}$$

In the parallel configuration, the projection matrices may be chosen so that $\mathbf{\Pi}_0 = [\mathbf{I} \mid -\mathbf{C}_0]$ and $\mathbf{\Pi}_1 = [\mathbf{I} \mid -\mathbf{C}_1]$, where $\mathbf{I}$ is the 3×3 identity matrix. Without loss of generality, we assume that $\mathbf{C}_0$ is at the world origin and $\overline{\mathbf{C}_0\mathbf{C}_1}$ is parallel to the world X-axis so that $\mathbf{C}_1 = [C_X\ 0\ 0]^T$. Let $\mathbf{p}_0$ and $\mathbf{p}_1$ be projections of a scene point $\mathbf{P} = [X\ Y\ Z\ 1]^T$ in the two views, respectively.

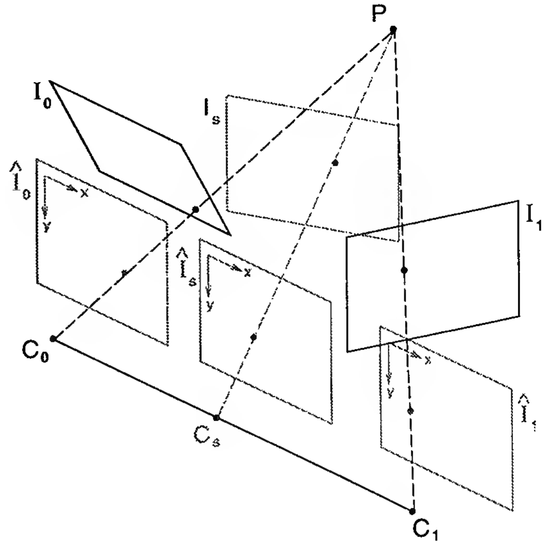


Figure 3: The three steps in view morphing: (1) Original images I_0 and I_1 are prewarped (rectified) to be parallel, (2) $\hat{I}_s$ is produced by interpolation, and (3) $\hat{I}_s$ is postwarped to form I_s.

Linear interpolation of $\mathbf{p}_0$ and $\mathbf{p}_1$ yields

$$\begin{aligned}(1-s)\mathbf{p}_0 + s\mathbf{p}_1 &= (1-s)\frac{1}{Z}\mathbf{\Pi}_0\mathbf{P} + s\frac{1}{Z}\mathbf{\Pi}_1\mathbf{P} \\ &= \frac{1}{Z}\mathbf{\Pi}_s\mathbf{P}\end{aligned}$$

where

$$\mathbf{\Pi}_s = (1-s)\mathbf{\Pi}_0 + s\mathbf{\Pi}_1 \qquad (1)$$

Image interpolation, or *morphing* [Beier and Neely, 1992], therefore produces a new view whose projection matrix, $\mathbf{\Pi}_s$, is a linear interpolation of $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$ and whose optical center is $\mathbf{C}_s = [sC_X\ 0\ 0]^T$. Eq. (1) indicates that in the parallel configuration, any parallel view along $\overline{\mathbf{C}_1\mathbf{C}_2}$ may be synthesized simply by interpolating corresponding points in the two basis views. In other words, image interpolation induces an interpolation of viewpoint for this special camera geometry.

To interpolate general views with projection matrices $\mathbf{\Pi}_0 = [\mathbf{H}_0 \mid -\mathbf{H}_0\mathbf{C}_0]$ and $\mathbf{\Pi}_1 = [\mathbf{H}_1 \mid -\mathbf{H}_1\mathbf{C}_1]$, we first apply homographies $\mathbf{H}_0^{-1}$ and $\mathbf{H}_1^{-1}$ to convert I_0 and I_1 to a parallel configuration. This procedure is identical to rectification techniques used in stereo vision

[Robert *et al.*, 1995]. This suggests a three-step procedure for view synthesis:

1. Prewarp: $\hat{I}_0 = \mathbf{H}_0^{-1} I_0$, $\hat{I}_1 = \mathbf{H}_1^{-1} I_1$
2. Morph: linearly interpolate positions and intensities of corresponding pixels in $\hat{I}_0$ and $\hat{I}_1$ to form $\hat{I}_s$
3. Postwarp: $I_s = \mathbf{H}_s \hat{I}_s$

Rectification is possible providing that the epipoles are outside of the respective image borders. If this condition is not satisfied, it is still possible to apply the procedure if the prewarped images are never explicitly constructed, i.e., if the prewarp, morph, and postwarp transforms are concatenated into a pair of aggregate warps [Seitz and Dyer, 1996b]. The prewarp step implicitly requires selection of a particular epipolar plane on which to reproject the basis images. Although the particular plane can be chosen arbitrarily, certain planes may be more suitable due to image sampling considerations.

4 Uncalibrated View Morphing

In order to use the view morphing algorithm presented in Section 3, we must find a way to rectify the images without knowing the projection matrices. Towards this end, it can be shown [Seitz and Dyer, 1996a] that two images are in the parallel configuration when their fundamental matrix is given, up to scalar multiplication, by

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

We seek a pair of homographies $\mathbf{H}_0$ and $\mathbf{H}_1$ such that the prewarped images $\hat{I}_0 = \mathbf{H}_0^{-1} I_0$ and $\hat{I}_1 = \mathbf{H}_1^{-1} I_1$ have the fundamental matrix given by Eq. (2). In terms of $\mathbf{F}$ the condition on $\mathbf{H}_0$ and $\mathbf{H}_1$ is

$$\mathbf{H}_1{}^T \mathbf{F} \mathbf{H}_0 = \hat{\mathbf{F}} \qquad (2)$$

Solutions to Eq. (2) are discussed in [Seitz and Dyer, 1996a, Robert *et al.*, 1995].

We have established that two images can be rectified, and therefore interpolated, without knowing their projection matrices. As in Section 3, interpolation of the prewarped images results in new views along $\overline{\mathbf{C}_0\mathbf{C}_1}$. In contrast to the calibrated case however, the postwarp step is underspecified; there is no obvious choice for the homography that transforms $\hat{I}_s$ to I_s. One solution is to have the user provide the homography directly or indirectly by specification of a small number of image points [Laveau and Faugeras, 1994, Seitz and Dyer, 1996b]. Another method is to simply interpolate the components of $\mathbf{H}_0^{-1}$ and $\mathbf{H}_1^{-1}$, resulting in a continuous transition from I_0 to I_1 [Seitz and Dyer, 1996a]. Both methods for choosing the postwarp transforms generally result in the synthesis of *projective* views. A projective view is a perspective view warped by a 2D affine transformation.

5 Three Views and Beyond

The paper up to this point has focused on image synthesis from exactly two basis views. The extension to more views is straightforward. Suppose for instance that we have three basis views that satisfy monotonicity pairwise ((I_0, I_1), (I_0, I_2), and (I_1, I_2) each satisfy monotonicity). Three basis views permit synthesis of a triangular region of viewspace, delimited by the three optical centers. Each pair of basis images determines the views along one side of the triangle, spanned by $\overline{\mathbf{C}_0\mathbf{C}_1}$, $\overline{\mathbf{C}_1\mathbf{C}_2}$, and $\overline{\mathbf{C}_2\mathbf{C}_0}$.

What about interior views, i.e., views with optical centers in the interior of the triangle? Indeed, any interior view can be synthesized by a second interpolation, between a corner and a side view of the triangle. However, the assumption that monotonicity applies pairwise between corner views is not sufficient to infer monotonicity between interior views in the closed triangle $\triangle\mathbf{C}_0\mathbf{C}_1\mathbf{C}_2$; monotonicity is not transitive. In order to predict interior views, a slightly stronger constraint is needed. *Strong monotonicity* dictates that for every pair of scene points $\mathbf{P}$ and $\mathbf{Q}$, the line $\mathbf{PQ}$ does not intersect $\triangle\mathbf{C}_0\mathbf{C}_1\mathbf{C}_2$. Strong monotonicity is a direct generalization of monotonicity; in particular, strong monotonicity of $\triangle\mathbf{C}_0\mathbf{C}_1\mathbf{C}_2$ implies that monotonicity is satisfied between every pair of views centered in this triangle, and vice-versa. Consequently, strong monotonicity permits synthesis of any view in $\triangle \mathbf{C}_0\mathbf{C}_1\mathbf{C}_2$.

Now suppose we have n basis views with optical centers $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1}$ and that strong monotonicity applies between each triplet of basis views[1]. By the preceding argument, any triplet of basis views determines the triangle of views between them. In particular, any view on the convex hull $\mathcal{H}$ of $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1}$ is determined, as $\mathcal{H}$ is comprised of a subset of these triangles. Furthermore, the interior views are also determined: let $\mathbf{C}$ be a point in the interior of $\mathcal{H}$ and choose a corner $\mathbf{C}_i$ on $\mathcal{H}$. The line through $\mathbf{C}$ and $\mathbf{C}_i$ intersects $\mathcal{H}$ in a point $\mathbf{K}$. Since $\mathbf{K}$ lies on the convex hull, it represents the optical center of a set of views produced by two or fewer interpolations. Because $\mathbf{C}$ lies on $\overline{\mathbf{C}_i\mathbf{K}}$, all views centered at $\mathbf{C}$ are determined as well by one additional interpolation, providing monotonicity is satisfied between $\mathbf{C}_i$ and $\mathbf{K}$. To establish this last condition, observe that for monotonicity to be violated there must exist two scene points $\mathbf{P}$ and $\mathbf{Q}$ such that $\mathbf{PQ}$ intersects $\overline{\mathbf{C}_i\mathbf{K}}$, implying that $\mathbf{PQ}$ also intersects $\mathcal{H}$. Thus, $\mathbf{PQ}$ intersects at least one triangle $\triangle \mathbf{C}_i\mathbf{C}_j\mathbf{C}_k$ on $\mathcal{H}$, violating the assumption of strong monotonicity. In conclusion, n basis views determine the 3D range of viewspace contained in the convex hull of their optical centers.

This constructive argument suggests that arbitrarily large regions of viewspace may be constructed by adding more basis views. However, the prediction of any range of view-space depends on the assumption that *all* possible pairs of views within that space satisfy monotonicity. In particular, a monotonic range may span no more than a single aspect of an aspect graph [Seitz and Dyer, 1996a], thus limiting the range of views that may be predicted. Nevertheless, it is clear that a discrete set of views implicitly describes scene appearance from a continuous range of viewpoints.

6 Experimental Results

We have applied the view morphing algorithm to many pairs of basis images, two of which are shown in Figure 4. Each pair of images was uncalibrated and the fundamental matrix was computed from several manually-specified point correspondences.

The first pair of images shows two views of a face. A sparse set of user-specified feature correspondences was used to determine the correspondence map [Seitz and Dyer, 1996b]. The synthesized image represents a view halfway between the two basis views. Some artifacts occur in regions where monotonicity is violated, e.g., near the right ear.

The second pair of images shows a wooden mannequin. This is an object that would be difficult to reconstruct due to lack of texture, but is relatively easy to synthesize views. In this example, image correspondences were automatically determined. Some local artifacts are visible where monotonicity is violated (e.g., left foot). Blurring is caused by image resampling, which is done three times in the current implementation. The problem may be ameliorated by super-sampling the intermediate images or by concatenating the multiple image transforms into two aggregate warps and resampling only once [Seitz and Dyer, 1996b].

7 Conclusions

In this paper we considered the question of which views of a static scene may be predicted from a set of two or more basis views, under perspective projection. The following results were shown: under monotonicity, two perspective views determine scene appearance from the set of all viewpoints on the line between their optical centers. Second, under strong monotonicity, a volume of viewspace is determined, corresponding to the convex hull of the optical centers of the basis views. Third, new perspective views may be synthesized by rectifying a pair of images and then interpolating corresponding pixels, one scanline at a time, using a procedure called *view morphing.* Fourth, view synthesis is possible even when the views are

[1] In fact, strong monotonicity for each triangle on the convex hull of $\mathbf{C}_0, \ldots, \mathbf{C}_{n-1}$ is sufficient.

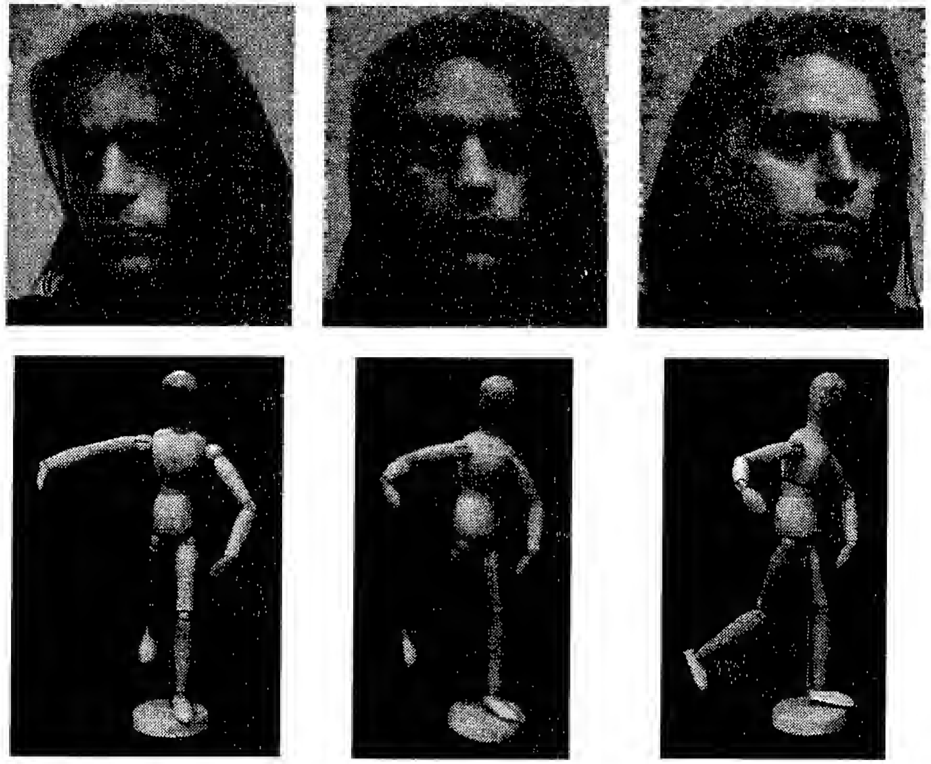


Figure 4: Basis views (left and right) of a face (top) and mannequin (bottom), with a synthesized view (center) halfway in-between each pair.

uncalibrated, provided the *fundamental matrix* is known. In the uncalibrated case, the synthesized images represent *projective* views of the scene.

References

[Beier and Neely, 1992] T. Beier and S. Neely. Feature-based image metamorphosis. In *Proc. SIGGRAPH 92*, pages 35–42, 1992.

[Beymer and Poggio, 1996] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272:1905–1909, 1996.

[Chen and Williams, 1993] S. Chen and L. Williams. View interpolation for image synthesis. In *Proc. SIGGRAPH 93*, pages 279–288, 1993.

[Laveau and Faugeras, 1994] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images. In *Proc. 12th Int. Conf. Pattern Recognition*, pages 689–691, 1994.

[McMillan and Bishop, 1995] L. McMillan and G. Bishop. Plenoptic modeling. In *Proc. SIGGRAPH 95*, pages 39–46, 1995.

[Robert *et al.*, 1995] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually guided robotics tasks. Technical Report 2584, INRIA, Sophia-Antipolis, France, June 1995.

[Seitz and Dyer, 1995] S. Seitz and C. Dyer. Physically-valid view synthesis by image interpolation. In *Proc. IEEE Workshop on Representation of Visual Scenes*, pages 18–25, 1995.

[Seitz and Dyer, 1996a] S. Seitz and C. Dyer. Scene appearance representation by perspective view synthesis. Technical Report 1298, University of Wisconsin, May 1996.

[Seitz and Dyer, 1996b] S. Seitz and C. Dyer. View morphing. In *Proc. SIGGRAPH 96*, pages 21–30, 1996.

Direct methods for estimation of structure and motion from three views

G. P. Stein

Artificial Intelligence Laboratory
MIT
Cambridge, MA 02139
gideon@ai.mit.edu

A. Shashua

Institute of Computer Science
Hebrew University of Jerusalem
Jerusalem 91904, Israel
http://www.cs.huji.ac.il/~ shashua/

Abstract

We describe a new 'direct method' for estimating structure and motion from image intensities of multiple views. We extend the direct methods of [5] to three views. Adding the third view enables us to solve for motion, and compute a dense depth map of the scene, directly from image spatio-temporal derivatives in a linear manner without first having to find point correspondences or compute optical flow.

We describe the advantages and limitations of this method and show experiments with real images.

1 Introduction

We present a new method for computing motion and dense structure from three views. This method can be viewed as an extension of the 'direct methods' of Horn and Weldon [5] from two views (one motion) to three views (two motions). These methods are dubbed 'direct methods' because they do not require prior computation of optical flow. We assume small image motions on the order of a few pixels.

Applying the constant brightness constraint [4] to the trilinear tensor of Shashua and Werman [9, 12] results in an equation relating camera motion and calibration parameters to the image gradients (first order only). We get one equation for each point in the image with a fixed number of parameters. This results in a highly over-constrained set of equations.

This method has advantages over both optical flow methods [6][7] and feature based methods [12]. We combine the information from all the points in the image and thus we avoid the aperture problem which makes computation of optical flow difficult. We do not explicitly define feature points. Points with small gradients simply contribute less to the least squares estimation. Information from all points that have gradients is used.

These advantages are highlighted in a scene with a set of vertical bars in front of a set of horizontal bars and behind everything a uniform background. Optical flow methods fail because of the straight bars and the aperture problem. Discrete methods fail because the intersections of the lines in the image, which are detected as 'features' do not correspond to real features in space. Many natural scenes such as tree branches or man made objects, window frames, lamp posts and fences give rise to these problems.

Starting with the general uncalibrated model we proceed through a hierarchy of reduced models first by assuming calibrated cameras and then by assuming the Longuett-Higgins and Prazdny small motion model [6]. We then show how to solve the simplified model for the motion parameters.

1.1 Previous Work

The 'direct methods' were pioneered by Horn and Weldon in [5]. A single image pair results in N equations in $N+6$ unknowns, where N is the number of points in the image, so a constraint is needed. Negahdaripour and Horn [8] present a closed form solution assuming a planar or quadratic surface.

This work is based on the work of Shashua and Hanna [11]. Here we describe the results of implementing these ideas in practice. During the course of implementation various subtleties and limitations were discovered.

2 Mathematical Background

2.1 Notations

Let $\boldsymbol{x}$ be a point in 3D space and its projection in a pair of images be p and p'. Then $p = [I; 0]\boldsymbol{x}$ and $p' \cong A\boldsymbol{x}$, the left 3×3 minor of A stands for a 2D projective transformation of the chosen plane at infinity and the fourth column of A stands for the epipole (the projection of the center of camera 0 on the image plane of camera 1). In particular, in a calibrated setting the 2D projective transformation is the rotational component of camera motion and the epipole is the translational component of camera

[0] G.S would like to acknowledge DARPA contracts N00014-94-01-0994 and 95009-5381. A.S. would like to acknowledge US-IS BSF contract 94-00120 and the European ACTS project AC074.

motion.

We will occasionally use tensorial notations as described next. We use the covariant-contravariant summation convention: a point is an object whose coordinates are specified with superscripts, i.e., $p^i = (p^1, p^2, ...)$. These are called contravariant vectors. An element in the dual space (representing hyperplanes — lines in $\mathcal{P}^2$), is called a covariant vector and is represented by subscripts, i.e., $s_j = (s_1, s_2, ...)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^i s_i = p^1 s_1 + p^2 s_2 + ... + p^n s_n$. This is known as a contraction.

Matching image points across three views will be denoted by p, p', p''; the coordinates will be referred to as p^i, p'^j, p''^k, or alternatively as non-homogeneous image coordinates $(x, y), (x', y'), (x'', y'')$.

2.2 Stratification of Motion Models

Three views, $p = [I; 0]\boldsymbol{x}, p' \cong A\boldsymbol{x}$ and $p'' \cong B\boldsymbol{x}$, are known to produce four trilinear forms whose coefficients are arranged in a tensor representing a bilinear function of the camera matrices A, B:

$$\alpha_i^{jk} = t'^j b_i^k - t''^k a_i^j \tag{1}$$

where $A = [a_i^j, t'^j]$ (a_i^j is the 3×3 left minor and t' is the fourth column of A) and $B = [b_i^k, t''^k]$. The tensor acts on a triplet of matching points in the following way:

$$p^i s_j^\mu r_k^\rho \alpha_i^{jk} = 0 \tag{2}$$

where s_j^μ are any two lines (s_j^1 and s_j^2) intersecting at p', and r_k^ρ are any two lines intersecting p''. Since the free indices are μ, ρ each in the range 1,2, we have 4 trilinear equations (unique up to linear combinations). More details can be found in [3, 9, 12, 10].

Geometrically, a trilinear matching constraint is produced by contracting the tensor with the point p of image 0, *some* line coincident with p' in image 1, and *some* line coincident with p'' in image 2. We can describe any line through p', as a linear combination of the vertical $(1, 0, -x')$ and horizontal $(0, 1, -y')$ lines. Let the coefficients of the linear combination be the components of the image gradient I_x, I_y at (x, y) in image 0, then the line s' has the form:

$$S' = \begin{pmatrix} I_x \\ I_y \\ -x'I_x - y'I_y \end{pmatrix}$$

The contribution of x', y' can be removed by using the constant brightness equation due to [4]:

$$u'I_x + v'I_y + I'_t = 0 \tag{3}$$

where $u' = x - x'$, $v' = y - y'$ and I'_t is the discrete temporal derivative at (x, y), i.e., $I_1(x, y) - I_0(x, y)$. I_1 and I_0 are the image intensity values of the second and first images, respectively. Following the substitution we obtain,

$$S' = \begin{pmatrix} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{pmatrix} \tag{4}$$

Likewise, for the third image we have a simillar expression with $I''_t - xI_x - yI_y$ replacing the third line in S'', where I''_t is the temporal derivative between images 0 and 2. The tensor brightness constraint is therefore:

$$\boxed{s''_k s'_j p^i \alpha_i^{jk} = 0.} \tag{5}$$

We get a constraint equation involving the unknowns α_i^{jk} and the spatio-temporal derivatives at each pixel — the constraint is linear in the unknowns. This constraint was introduced by [11]. In other words, one can recover in principle the camera matrices across three views in the context of the "aperture" problem, as noticed by [13].

Starting from the general model (27 parameter model) of the constraint equation one can introduce a hierarchy of reduced models, as follows. By enforcing small-angle rotation on the camera motions, i.e., $A = [I + [w']_x; t']$ and $B = [I + [w'']_x; t'']$ where w', w'' are the angular velocity vectors and $[\cdot]_x$ is the skew-symmetric matrix of vector products, the tensor brightness constraint is reduced to a 24-parameter model which in matrix form looks like:

$$I''_t S'^{\top} t' - I'_t S''^{\top} t'' + S'^{\top}[t' w''^{\top}]V'' - S''^{\top}[t'' w'^{\top}]V' = 0, \tag{6}$$

where $V' = p \times S'$ and $V'' = p \times S''$. If, in addition, we enforce infinitesimal translational motion (the Longuett-Higgins & Prazdny [6] motion model), which results in the image motion equations:

$$u' = \frac{1}{z}(t'_1 - xt'_3) - w'_3 y + w'_2(1 + x^2) - w'_1 xy \tag{7}$$
$$v' = \frac{1}{z}(t'_2 - yt'_3) + w'_3 x - w'_1(1 + y^2) + w'_2 xy$$

then S has the simpler form:

$$S = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix} \tag{8}$$

and

$$V = p \times S = \begin{pmatrix} -I_y - y(xI_x + yI_y) \\ I_x + x(xI_x + yI_y) \\ xI_y - yI_x \end{pmatrix} \tag{9}$$

We obtain a 15-parameter model of the form:

$$I''_t S^T t' - I'_t S^T t'' + S^T [t' w''^T - t'' w'^T] V = 0 \tag{10}$$

We have one such equation for each point in the image. It is a set of bilinear equations in the unknowns t', t'', w', w''. After solving for the camera motions (to be described later in the paper) we can solve for the dense depth map from the equations:

$$KS^{\top} t' + V^{\top} w' + I'_t = 0 \tag{11}$$

$$KS^{\top} t'' + V^{\top} w'' + I''_t = 0 \tag{12}$$

where $K = \frac{1}{z}$ denotes inverse of the depth at each pixel location. Equations (11)(12) were introduced in [5] and can be obtained by substituting (eq. 7) in equation (3) and rearranging the terms.

3 Solving the bilinear equation

3.1 The pure translation case

In the pure translation case equation (10) becomes:

$$I_t'' S^T t' - I_t' S^T t'' = 0 \tag{13}$$

We have one such equation for each image point. Writen in matrix form this becomes $At = 0$ where $t = (t', t'')^T$ and A is an $N \times 6$ matrix with the $i'th$ row (corresponding to the $i'th$ pixel) given by:

$$\begin{pmatrix} I_t'' S_{i1} & I_t'' S_{i2} & I_t'' S_{i3} & I_t' S_{i1} & I_t' S_{i2} & I_t' S_{i3} \end{pmatrix} \tag{14}$$

We avoid the trivial solution $t = 0$ by adding the constraint $||t|| = 1$. The least squares problem now maps to problem of finding $||t|| = 1$ that minimizes:

$$t^T A^T A t = 0 \tag{15}$$

The solution is the eigenvector of $A^T A$ corresponding to the smallest eigenvalue.

3.2 Translation with rotation

In the general Longuett-Higgins and Prazdny model we are confronted with the bilinear equation (10). We treat the 6 translation parameters and the 9 outer product terms $[t' w''^T - t'' w'^T]$ as 15 intermediate parameters which are solved for as in section (3.1) but with a 15×15 matrix A. After recovering the 15 intermediate parameters we compute w' and w'' from $[t' w''^T - t'' w'^T]$.

There is one problem. If we consider the $N \times 15$ matrix A we note that the vector

$$c_0 = (0,0,0,0,0,0,1,0,0,0,1,0,0,0,1)^T \tag{16}$$

is in the null space of A. We note that c_0 is an inadmissable solution. The correct solution vector is also in the null space of A. Therefore the null space of A has *rank* ≥ 2 and A is of *rank* ≤ 13.

Suppose we found a vector b_0 in the null space of A such that $c_0 \times b_0 \neq 0$. The two vectors c_0 and b_0 span the null space of A and the desired solution vector b is a linear combination of the two:

$$b = b_0 + \alpha c_0 \tag{17}$$

In order to find α given c_0 and b_0 we must apply some constraint. We choose to enforce the constraint that the matrix $B = [t' w''^T - t'' w'^T]$ be of rank 2. Let B_0 and C_0 be the 9 last elements of the vectors b_0 and c_0 organized in 3×3 matrices. Since $C_0 = I$, B will be of rank 2 if α is an eigenvalue of B_0. This gives up to 3 solutions and we chose the eigenvalue with the smallest absolute value.

4 Implementation details

4.1 Iterative refinement and coarse to fine processing

The constant brightness constraint is a linearized form of the Sum Square Difference (SSD) criteria. The linear solution is therefore a single iteration of Newton's method. For iterative refinement, first one calculates motion and depth using the above equations. Then, using the depth and motion, images 1 and 2 are warped towards image 0. A correction to the motion and depth estimates are computed using the warped images. In the ideal case, as the final result, the warped images should appear nearly identical to image 0. For details see [14].

In order to deal with image motions larger than 1 pixel we use a Gaussian pyramid for coarse to fine processing [1][2].

4.2 Computing the depth, smoothing and interpolation.

After recovering camera motions, using equations (11) and (12) we compute depth at every point where $S^T t' \neq 0$ or $S^T t'' \neq 0$. We combine information from both images and interpolate over areas where image gradients are small using Local Weighted Regression. This could be replaced by other methods such as RBF's, B-splines or thin-plate interpolation.

Equation (18) shows the cost function used to compute the depth at a given point:

$$\underset{K}{\text{min arg}} \sum_{x,y \in R} \sum_{j} \beta(x,y) |S^T t^j|^p \left(K S^T t^j + V^T w^j + I_t^j\right)^2 \tag{18}$$

We sum over a region R using a windowing function $\beta(x,y)$ and over the two motions $j = 1, \ldots, 2$. The $|S^T t^j|^p$ term reduces the weight of points with a small gradient or where the gradient is perpendicular to the motion. We used $p = 1$. During the iteration process we used a region (R) of 5×5 or 7×7.

5 Experiments with real images

5.1 Experimental procedure

Images were taken with a Phillips $\frac{1}{3}$*inch* CCD video camera and an 8*mm* lens using the SGI Indy built in frame grabber at 640×480 pixel resolution. Figure (1a) shows one of three images. Depth ranged from 450*mm* to 750*mm*. Camera motions were 3*mm* vertical and (about) 5*mm* horizontal. No special care was taken to ensure precise motion. Image motions were $6 - 11 pixels$.

5.2 Results

We used 4 levels of coarse to fine processing with 2 iterations at each level. Varying the number of iterations from 1 through 4 had no qualitative impact but using a single iteration caused a small change in the resulting motion estimates. A 7×7 region was used for the local constant depth fit at all levels. Table (1) shows the estimated camera motions. It shows that the first motion was along the Y axis and the second motion along the X axis and that for both cases the rotation was negligible. This is qualitatively correct. We do not have accurate ground truth estimates.

Figure (1b) shows the recovered depth map (in fact this shows $K(x,y) = \frac{1}{Z}$). Figure (1c) shows a 3D

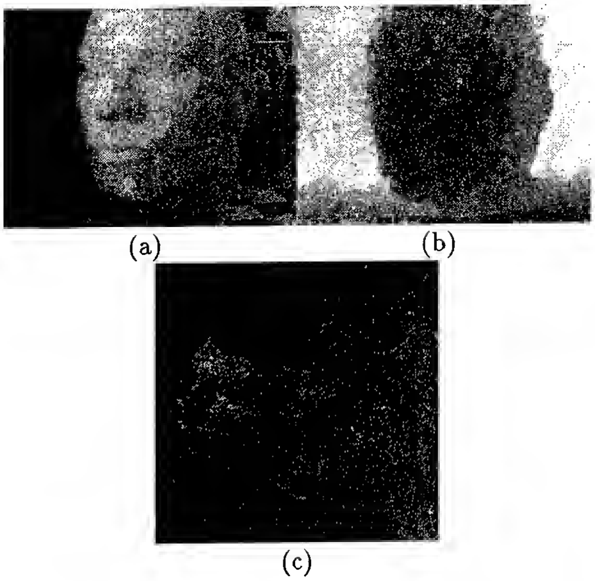


Figure 1: *One of the three input images (a) and the estimated $K(x,y) = \frac{1}{Z}$ inverse depth map (b) and 3D rendering of the surface with K scaled by 100.0 (c). Uses 7×7 region and a local constant depth model.*

rendering of the surface $K(x,y)$. The K values have been scaled by 100.0. In order to get smoother and more visually pleasing results a local planar model ($K(x,y) = K_x x + K_y y + K_0$) was used for the final stage with a 30×30 region of support. The results are shown in figures (2a, 2b). There is noticeable smoothing and overshoots at depth discontinuities and the tip of the nose. In (2b) the texture was removed for clarity.

6 Discussion and future work

We have presented a new method for recovering structure and motion from 3 views which does not require feature correspondence or optical flow. We have shown, using real images, that the method can qualitatively recover depth and motion in the general, small motion case. These results are promising but more experiments are needed to test the accuracy of the motion estimation.

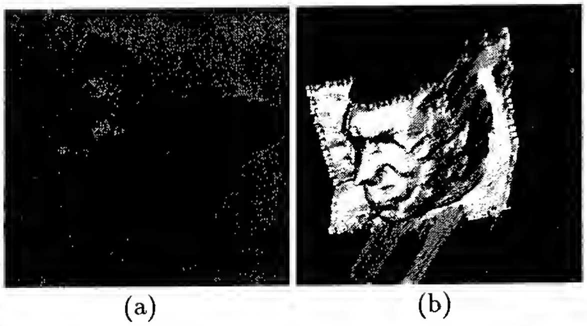


Figure 2: *3D rendering of the surface $K(x,y) = \frac{1}{Z}$. Uses a 30×30 region and a locally planar depth model. Note the overshoot at depth discontinuities around the head.*

Table 1: Motion estimates from real images.

FOE 1	(-603.7, -8055)
FOE 2	(16302, 300.2)
W1	(0.00022, -0.00055, -0.0217)
W2	(-0.00027 -0.00017 -0.00062)

References

[1] Bergen, J.R., Anandan, P., Hanna, K.J. and Hingorani, R., "Hierarchical model-based motion estimation", In *Proceedings EECV*, Santa Margherita Ligure, Italy, June (1992)

[2] Burt, P.J. and Adelson, E.H., "The Laplacian pyramid as a compact image code" *IEEE Transactions on Communications*, 31:532-540, (1983)

[3] Hartley, R., "A linear method for reconstruction from lines and points" In *Proceedings ICCV*, 882-887, Cambridge, MA, June (1995)

[4] Horn, B.K.P. and Schunk, B.G., "Determining optical flow" *Artificial Intelligence*, 17:185-203 (1981)

[5] Horn, B.K.P. and Weldon, E.J., "Direct methods for recovering motion" *IJCV* 2:51-76, (1988)

[6] Longuett-Higgins, H.C. and Prazdny, K., "The interpretation of a moving retinal image." *Proceedings of the Royal Society of London B*, 208:385-397,(1980)

[7] Lucas, B.D. and Kanade, T., "An iterative image registration technique with an application to stereo vision" In *Proceedings IJCAI*, 674-679, Vancouver, (1981)

[8] Negahdaripour, S. and Horn, B.K.P, "Direct passive navigation". *PAMI*,9(1):168-176,(1987)

[9] Shashua, A., "Algebraic functions for recognition", *PAMI*, 17(8):779–789, 1995.

[10] Shashua, A. and Anandan, P. "Trilinear Constraints Revisited: Generalized Trilinear Constraints and the Tensor Brightness Constraint", *Proceedings of the ARPA IU Workshop*, Feb. 1996, Palm Springs, CA.

[11] Shashua, A. and Hanna, K.J., "The Tensor Brightness Constraints: Direct Estimation of Motion Revisited" Technion Technical Report, Haifa, Israel, November (1995)

[12] Shashua, A. and Werman, M., "Trilinearity of Three Perspective Views and its Associated Tensor", In *Proceedings of ICCV 95*, 920-925 Boston, MA, USA, June (1995)

[13] Spetsakis, M.E. and Aloimonos, J. "A unified theory of structure from motion", In *Proceedings IU Workshop*, 1990.

[14] Stein, G.P. and Shashua, A., "Direct methods for estimation of structure and motion from three views" AI Memo-1594, MIT, AI Lab (1996).

Dense Depth Maps from Epipolar Images

J.P. Mellor * **Seth Teller** **Tomás Lozano-Pérez**
MIT Artificial Intelligence Laboratory
MIT Computer Graphics Laboratory
545 Technology Square NE43-753
Cambridge, MA 02139
jpmellor@ai.mit.edu, http://www.graphics.lcs.mit.edu

Abstract

This paper describes a method for generating dense depth maps given large numbers of images taken from arbitrary positions. The algorithm presented is completely local and uses an epipolar image to generate for each pixel an evidence versus depth and surface normal distribution. In many cases, the distribution contains a clear and distinct global maximum. The location of this peak determines the depth and its shape can be used to estimate the error. The distribution can also be used to perform a maximum likelihood fit of models directly to the images. We anticipate that the ability to perform maximum likelihood estimation from purely local calculations will prove useful in constructing three dimensional models from large sets of images.

1 Introduction

One approach to improving the results obtained by stereo techniques is to use multiple images. Several researchers, such as Yachida [1986], have proposed trinocular stereo algorithms. Others have also used special camera configurations to aid in the correspondence problem [Tsai, 1983, Bolles *et al.*, 1987, Okutomi and Kanade, 1993]. The work presented here also uses multiple images and draws its major inspiration from Bolles, Baker and Marimont [1987]. We define a construct called an *epipolar image* and use it to analyze evidence about depth. Like Tsai [1983] and Okutomi and Kanade [1993] we define a cost function that is applied across multiple images, and like Cox [1996] we model the occlusion process. There are several important differences, however. The epipolar image we define is valid for arbitrary camera positions and models some forms of occlusion. Our method is intended to recover dense depth maps of built geometry (architectural facades) using thousands of images acquired from within the scene. In most cases, depth can be recovered using purely local information, avoiding the computational costs of global constraints. Where depth cannot be recovered using purely local information, the depth evidence from the epipolar image provides a principled distribution for use in a maximum-likelihood approach [Duda and Hart, 1973].

2 Our Approach

We assume that camera pose is known in an absolute coordinate system. Although relative positions are sufficient for the discussion in this section, global positions allow us to

*This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of the Defense under Office of Naval Research contract N00014-91-J-4038. The author was also supported by the Advanced Research Projects Agency of the Department of Defense under Rome Laboratory contract F3060-94-C-0204.

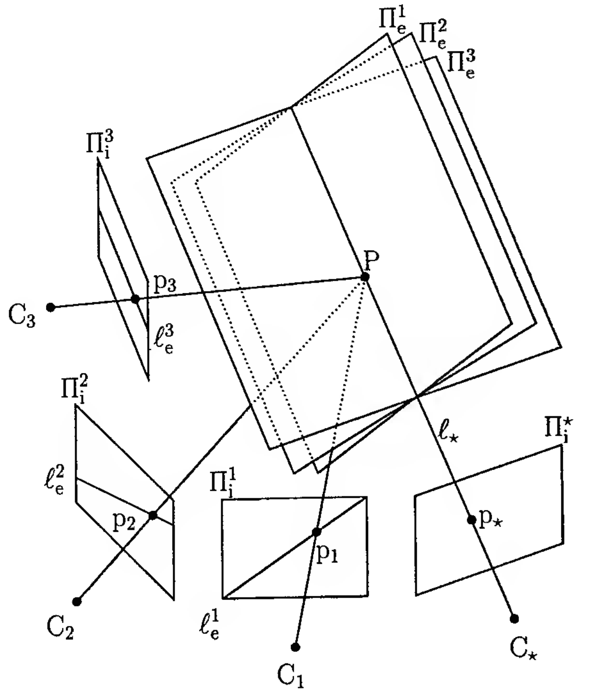


Figure 1: Epipolar image geometry.

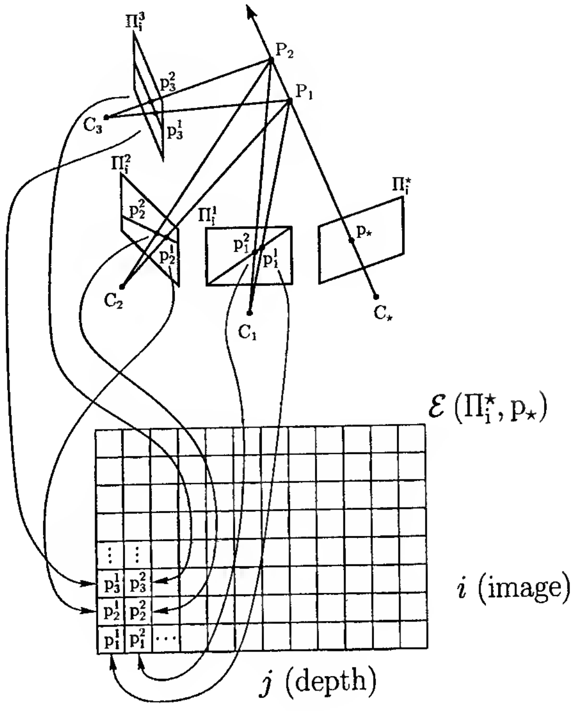


Figure 2: Constructing an epipolar image.

perform reconstruction incrementally using disjoint scenes. We also assume known internal camera parameters. For a more complete description of our method see [Mellor *et al.*, 1996].

2.1 Epipolar Images

For our analysis we will define an epipolar image $\mathcal{E}$ which is a function of one image and a point in that image. An epipolar image is similar to an epipolar-plane image [Bolles *et al.*, 1987], but has one critical difference that ensures it can be constructed for *every* pixel in an arbitrary set of images. Rather than use projections of a single epipolar plane, we construct the epipolar image from the *pencil* of epipolar planes defined by the line $\ell_\star$ through one of the camera centers $C_\star$ and one of the pixels $p_\star$ in that image[1] $\Pi_i^\star$ (Figure 1). Π_e^i is the epipolar plane formed by $\ell_\star$ and the i^{th} camera center C_i. Epipolar line ℓ_e^i contains all of the information about $\ell_\star$ present in Π_i^i.

To simplify the analysis of an epipolar image we can group points from the epipolar lines according to possible correspondences (Figure 2). P_1 projects to p_i^1 in Π_i^i; therefore $\{p_i^1\}$ has all of the information contained in $\{\Pi_i^i\}$ about P_1. Similarly, there is a distinct set for P_2; thus $\left\{p_i^j \mid \text{for a given } j\right\}$ contains all of the possible correspondences for P_j. If P_j is a point on the surface of a physical object and it is visible in $\{\Pi_i^i\}$ and $\Pi_i^\star$, then measurements taken at p_i^j should match[2] those taken at $p_\star$ (Figure 3a). Conversely, if P_j is not a point on the surface of a physical object then the measurements taken at p_i^j are unlikely to match those taken at $p_\star$ (Figures 3b and 3c). Epipolar images can be viewed as tables which accumulate evidence about possible correspondences of $p_\star$. A simple function of j is used to build $\left\{P_j \mid \forall i < j : \|P_i - C_\star\|^2 < \|P_j - C_\star\|^2\right\}$. In essence, $\{P_j\}$ is a set of samples along $\ell_\star$ at increasing depths from the image plane of $p_\star$.

[1] Π is used to denote a plane; the subscript identifies the type (epipolar or image); and the superscript identifies the instance.

[2] So far we have considered only diffuse surfaces. The matching function can be extended to account for specularity and we intend to do so in the future.

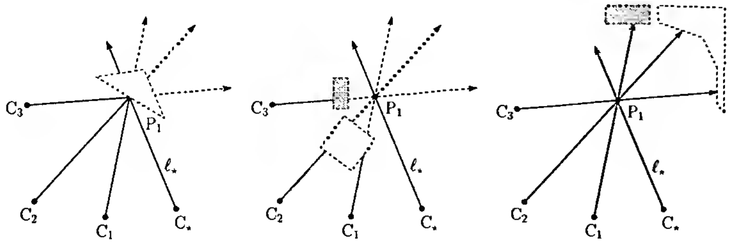


Figure 3: Typical cases.

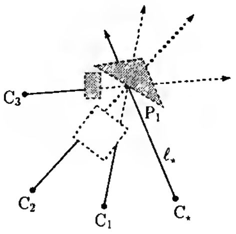


Figure 4: False negative.

2.2 Analyzing Epipolar Images

An epipolar image $\mathcal{E}$ is constructed by organizing image measurements into a two-dimensional array with i and j as the vertical and horizontal axes respectively. Rows in $\mathcal{E}$ are epipolar lines from different images; columns form sets of possible correspondences ordered by depth[3] (Figure 2). The quality $\nu(j)$ of the match between column j and $\mathrm{p}_\star$ can be thought of as evidence that $\mathrm{p}_\star$ is the projection of P_j with depth j. Real cameras have a finite field of view, and p_i^j may not be contained in the image Π_{i}^i $\left(\mathrm{p}_i^j \notin \{\Pi_{\mathrm{i}}^i\}\right)$. Thus, only terms for which $\mathrm{p}_i^j \in \{\Pi_{\mathrm{i}}^i\}$ should be included, producing

$$\nu(j) = \frac{\sum\limits_{i\,|\,\mathrm{p}_i^j \in \{\Pi_{\mathrm{i}}^i\}} \mathcal{X}(\mathcal{F}(\mathrm{p}_i^j), \mathcal{F}(\mathrm{p}_\star))}{\sum\limits_{i\,|\,\mathrm{p}_i^j \in \{\Pi_{\mathrm{i}}^i\}} 1} \qquad (1)$$

where $\mathcal{F}()$ is an image measurement and $\mathcal{X}()$ is a cost function which measures the difference between $\mathcal{F}(\mathrm{p}_i^j)$ and $\mathcal{F}(\mathrm{p}_\star)$. Ideally, $\nu(j)$ will have a sharp, distinct peak at the correct depth, so that

$$\arg\max_j(\nu(j)) = \text{ the correct depth of } \mathrm{p}_\star.$$

As the number of elements in $\left\{\mathrm{p}_i^j \,|\, \text{for a given } j\right\}$ increases, the likelihood increases that $\nu(j)$ will be large when P_j lies on a physical surface and small when it does not. Occlusion does not produce a peak at an incorrect depth or a false positive[4]. It can however, cause false negatives or the absence of a peak at the correct depth (Figure 4). A false negative is essentially a lack of evidence about the correct depth and can be addressed in two ways: removing the contribution of occluded views, and adding unoccluded views by acquiring more images.

A large class of occluded views can be eliminated quite simply. Each point P_j has an associated normal n_j. Images with camera centers in the negative half space defined by n_j cannot possibly have imaged P_j. Of course, n_j is not known a priori, but the fact that P_j is visible in $\Pi_{\mathrm{i}}^\star$ limits its possible values. This range of values can then be sampled and used to eliminate the contribution of occluded views from $\nu(j)$. Let α be an estimate of n_j and $\widehat{\mathrm{C}_i\mathrm{P}_j}$ be the unit vector along the ray from C_i to P_j, then P_j can only be visible if $\widehat{\mathrm{C}_i\mathrm{P}_j}\cdot\alpha < 0$. The updated function becomes:

$$\nu(j,\alpha) = \frac{\sum\limits_{i\in\mathcal{S}}\left(\widehat{\mathrm{C}_i\mathrm{P}_j}\cdot\alpha\right)\mathcal{X}(\mathcal{F}(\mathrm{p}_i^j), \mathcal{F}(\mathrm{p}_\star))}{\sum\limits_{i\in\mathcal{S}}\widehat{\mathrm{C}_i\mathrm{P}_j}\cdot\alpha} \qquad (2)$$

where

$$\mathcal{S} = \left\{ i \;\middle|\; \begin{array}{l} \mathrm{p}_i^j \in \{\Pi_{\mathrm{i}}^i\} \\ \widehat{\mathrm{C}_i\mathrm{P}_j}\cdot\alpha < 0 \end{array} \right\}.$$

Then, if sufficient evidence exists,

$$\arg\max_{j,\alpha}(\nu(j,\alpha)) \;\Rightarrow\; \left\{ \begin{array}{l} j = \text{ depth of } \mathrm{p}_\star \\ \alpha \text{ an estimate of } \mathrm{n}_j \end{array} \right. .$$

3 Results

Synthetic imagery was used to explore the characteristics of $\nu(j)$ and $\nu(j,\alpha)$. A CAD model of Technology Square, the four-building complex housing our laboratory, was

[3] The depth of P_j can be trivially calculated from j, therefore we consider j and depth to be interchangeable.

[4] Except possibly in an adversarial setting.

built by hand. The locations and geometries of the buildings were determined using traditional survey techniques. Photographs of the buildings were used to extract texture maps which were matched with the survey data. This three-dimensional model was then rendered from 100 positions along a "walk around the block". From this set of images, a $\Pi_{\mathrm{i}}^{\star}$ and $\mathrm{p}_{\star}$ were chosen and an epipolar image $\mathcal{E}$ constructed. $\mathcal{E}$ was then analyzed using equations 1 and 2 where[5]

$$\mathcal{F}(x) = \mathrm{hsv}(x) = [\mathrm{h}(x), \mathrm{s}(x), \mathrm{v}(x)]^{\mathrm{T}} \tag{3}$$

and

$$\begin{aligned} &\mathcal{X}([h_1, s_1, v_1]^{\mathrm{T}}, [h_2, s_2, v_2]^{\mathrm{T}}) = \\ &\quad -\left(\frac{s_1 + s_2}{2}\right)(1 - \cos(h_1 - h_2)) - \\ &\quad (2 - s_1 - s_2)\,|v_1 - v_2|\,. \end{aligned} \tag{4}$$

Figure 5 shows a base image $\Pi_{\mathrm{i}}^{\star}$ with $\mathrm{p}_{\star}$ marked by a cross. Under $\Pi_{\mathrm{i}}^{\star}$ is the epipolar image $\mathcal{E}$ generated using the remaining 99 images. Below $\mathcal{E}$ is the matching function $\nu(j)$ (1) and $\nu(j, \alpha)$ (2). The horizontal scale, j or depth, is the same for $\mathcal{E}$, $\nu(j)$ and $\nu(j, \alpha)$. The vertical axis of $\mathcal{E}$ is the image index, and of $\nu(j, \alpha)$ is a coarse estimate of the orientation α at P_j. The vertical axis of $\nu(j)$ has no significance; it is a single row that has been replicated for clarity. To the right, $\nu(j)$ and $\nu(j, \alpha)$ are also shown as two-dimensional plots[6].

Figure 5a shows the epipolar image that results when the upper left-hand corner of the foreground building is chosen as $\mathrm{p}_{\star}$. Near the bottom of $\mathcal{E}$, ℓ_{e}^{i} is close to horizontal, and p_i^j is the projection of blue sky everywhere except at the building corner. The corner points show up in $\mathcal{E}$ near the right side as a vertical streak. This is as expected since the construction of $\mathcal{E}$ places the projections of P_j in the same column. Near the middle of $\mathcal{E}$, the long horizontal streaks result because P_j is occluded, and near the top the large black region is produced because $\mathrm{p}_i^j \notin \Pi_{\mathrm{i}}^{i}$. Both $\nu(j)$ and $\nu(j, \alpha)$ have a sharp peak[7] that corresponds to the vertical stack of corner points. This peak occurs at a depth of 2375 units ($j = 321$) for $\nu(j)$ and a depth of 2385 ($j = 322$) for $\nu(j, \alpha)$. The actual distance to the corner is 2387.4 units. The reconstructed world coordinates of $\mathrm{p}_{\star}$ are $[-1441, -3084, 1830]^{\mathrm{T}}$ and $[-1438, -3077, 1837]^{\mathrm{T}}$ respectively. The actual coordinates[8] are $[-1446, -3078, 1846]^{\mathrm{T}}$.

In Figure 5b, $\mathrm{p}_{\star}$ is a point from the interior of a building face with highly periodic texture. There is a clear peak in $\nu(j, \alpha)$ that agrees well with manual measurements and is better than that in $\nu(j)$. In Figure 5c, $\mathrm{p}_{\star}$ is a point on a building face that is occluded (Figure 4) in a number of views. Both $\nu(j)$ and $\nu(j, \alpha)$ produce fairly good peaks that agree with manual measurements.

To further test our method, we reconstructed the depth of a region in one of the images (Figure 6). For each pixel inside the black rectangle the global maximum of $\nu(j, \alpha)$ was taken as the depth of that pixel. Figure 7 shows the reconstructed world coordinates[9] for each of the 3000 pixels in the region. The cluster of points beyond the left end (near [0,2]) and at the right end of the building correspond to sky points. The actual world coordinates were calculated from the CAD model and are shown in grey. The camera position is marked by a grey line extending from the center of projection in the direction of the optical axis. The reconstruction was performed purely locally at each pixel. Global constraints such as ordering or smoothness were not imposed, and no attempt was made to remove depths with low confidence or otherwise post-process the global maximum of $\nu(j, \alpha)$.

[5] The well known hue, saturation, value color model is denoted by hsv.

[6] Actually, $\sum_{\alpha} \nu(j, \alpha) / \sum_{\alpha} 1$ is plotted for $\nu(j, \alpha)$.

[7] White indicates minimum error, black maximum.

[8] Some of the difference may be due to the fact that $\mathrm{p}_{\star}$ was chosen by hand and might not be the exact projection of the corner.

[9] All coordinates have been divided by 1000 to simplify the plots.

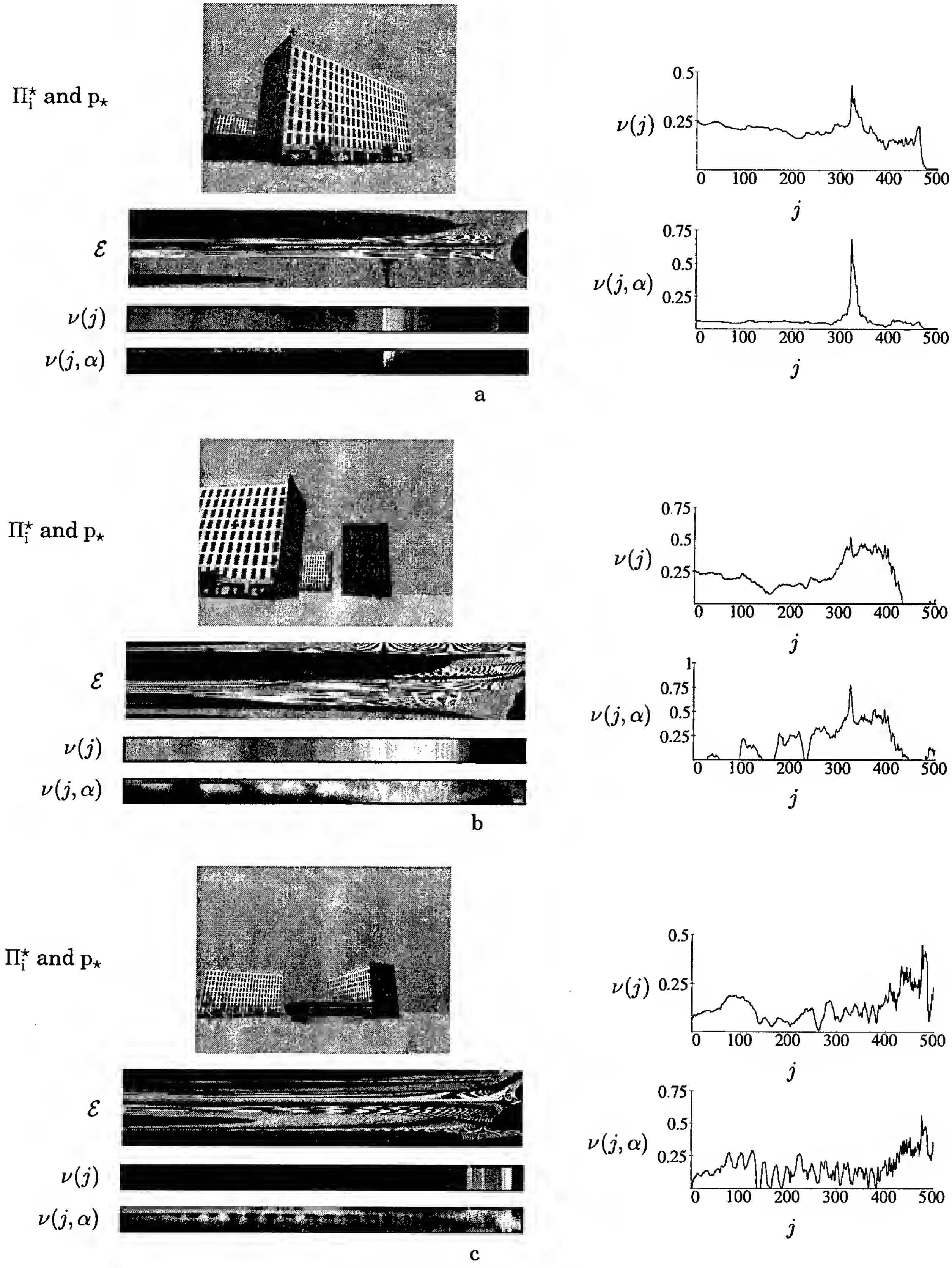


Figure 5: $\Pi_i^\star$, $p_\star$, $\mathcal{E}$, $\nu(j)$ and $\nu(j,\alpha)$.

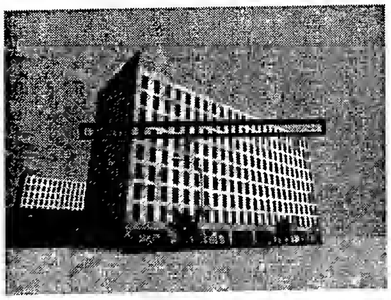
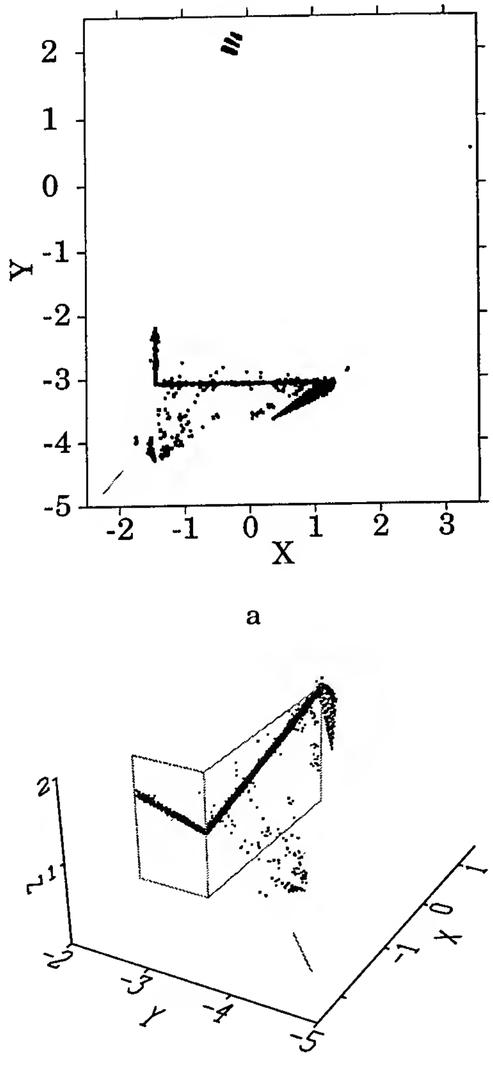


Figure 6: Reconstructed region.



a

b

Figure 7: Reconstructed and actual points.

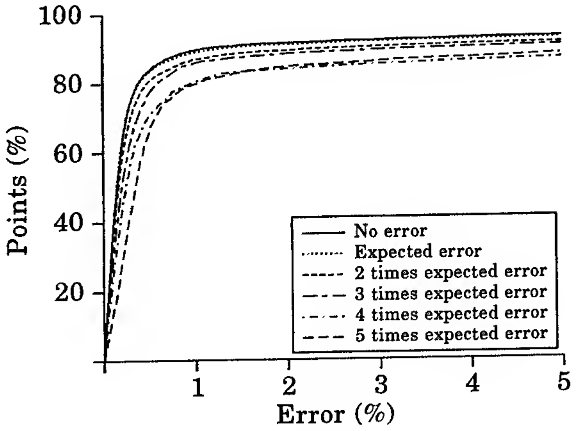


Figure 8: Number of points versus error.

Next, we considered outliers. Figure 8 shows the cumulative distribution of reconstruction error. Error is expressed as a percentage of the distance between the reconstructed $\mathrm{P_r}$ and actual $\mathrm{P_a}$ position divided by the depth of the reconstructed point ($\|\mathrm{P_r} - \mathrm{P_a}\| \,/\, \|\mathrm{P_r} - \mathrm{C}_\star\|$). The plotted curve indicates that the percentage[10] of reconstructed points with an error of less than 1% is 90% for the noise free case and 80% for noise levels of five times expected. Outliers also tend to have less support (fewer cameras contributing to the solution) and poor match quality (smaller values for $\nu(j, \alpha)$). Figure 9 shows the result of considering only points which have at least n cameras contributing to the solution. Similar results are obtained when points with small $\nu(j, \alpha)$ are removed.

Finally, we reconstructed another region about twice the size of the previous one which contained only building points. This time, we retained only points with more than 6 cameras contributing or with $\nu(j, \alpha) > -0.5$. Figure 10 shows the reconstructed points[11] rendered as oriented rectangular surface elements or *surfels* [Szeliski and Tonnesen, 1992]. We anticipate that the estimated orientation will prove very useful in fitting models to the reconstructed points or grouping them into surfaces.

[10] Sky points are omitted.

[11] Actually the data is downsampled by three in each direction for clarity.

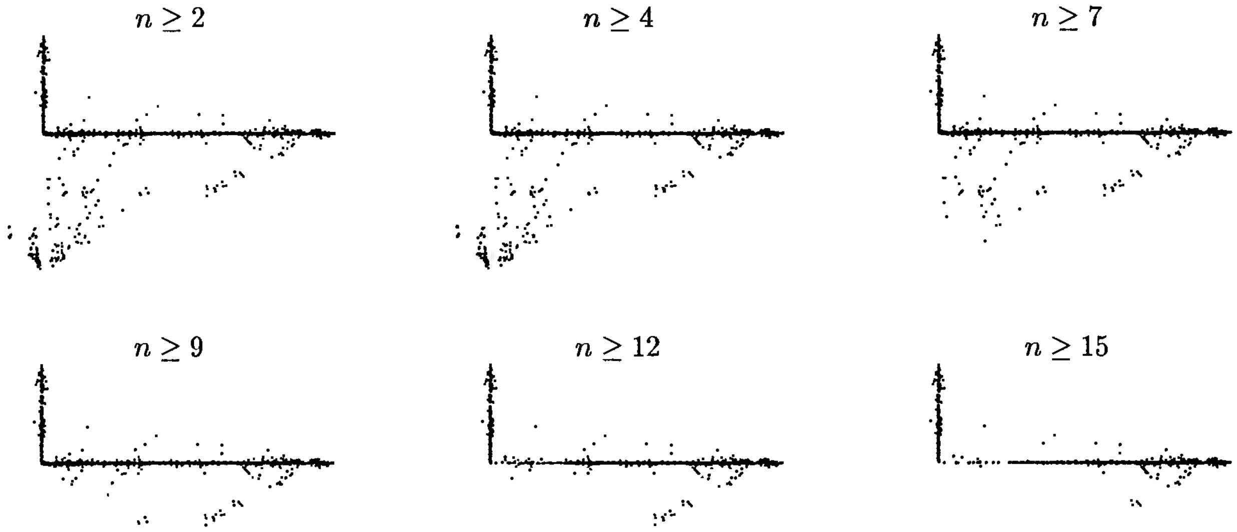


Figure 9: Outliers versus number of contributing cameras.

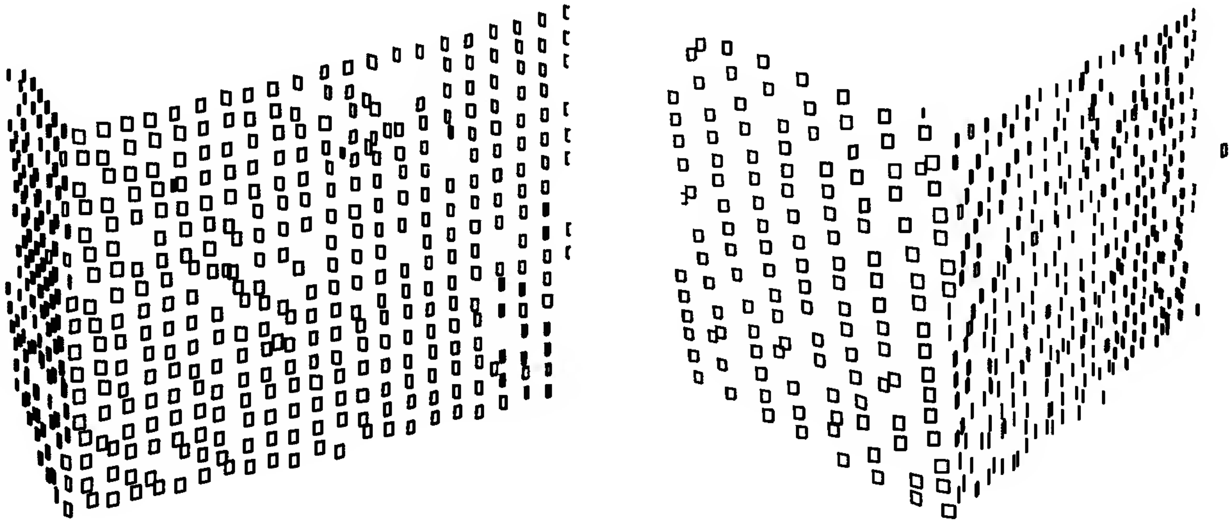


Figure 10: Two views of the reconstructed surfels.

4 Conclusions

This paper describes a method for generating dense depth maps directly from large sets of images taken from arbitrary poses. The algorithm presented is simple, accurate, and uses only local calculations. Our method builds, then analyzes, an epipolar image to accumulate evidence about the depth at each image pixel. This analysis produces an evidence versus depth and surface normal distribution that in many cases contains a clear and distinct global maximum. The location of this peak determines depth and orientation, and its shape can be used to estimate the error. The distribution can also be used to perform a maximum likelihood fit of models directly to the images. We anticipate that the ability to perform maximum likelihood estimation from purely local calculations will prove extremely useful in constructing three-dimensional models from large sets of images.

References

[Bolles *et al.*, 1987] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[Cox *et al.*, 1996] Ingemar J. Cox, Sunita L. Hingorani, Satish B. Rao, and Bruce M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[Duda and Hart, 1973] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

[Mellor *et al.*, 1996] J.P. Mellor, Seth Teller, and Tomás Lozano-Pérez. Dense depth maps from epipolar images. Technical Report AIM–1593, MIT, November 1996.

[Okutomi and Kanade, 1993] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.

[Szeliski and Tonnesen, 1992] Richard Szeliski and David Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 185–194, July 1992.

[Tsai, 1983] Roger Y. Tsai. Multiframe image point matching and 3-D surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):159–174, March 1983.

[Yachida, 1986] M. Yachida. 3D data acquisition by multiple views. In O. D. Faugeras and G. Giralt, editors, *Robotics Research: the Third International Symposium*, pages 11–18. MIT Press, Cambridge, MA, 1986.

Geometric Constraint Analysis and Synthesis: Methods for Improving Shape-Based Registration Accuracy

David A. Simon and Takeo Kanade
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213

Shape-based registration is a process for estimating the transformation between two shape representations of an object. It is used in many image-guided surgical systems to establish a transformation between pre- and intra-operative coordinate systems. This paper describes several tools which are useful for improving the accuracy resulting from shape-based registration: constraint analysis, constraint synthesis, and online accuracy estimation. Constraint analysis provides a scalar measure of sensitivity which is well correlated with registration accuracy. This measure can be used as a criterion function by constraint synthesis, an optimization process which generates configurations of registration data which maximize expected accuracy. Online accuracy estimation uses a conventional root-mean-squared error measure coupled with constraint analysis to estimate an upper bound on true registration error. This paper demonstrates that registration accuracy can be significantly improved via application of these methods.

1. Introduction

The registration process is a fundamental component of most image-guided surgical systems. Registration estimates a spatial transformation between two coordinate systems: a pre-operative system used to construct plans or simulations based upon medical data (e.g., CT, MRI, or X-ray images), and an intra-operative system in which the surgical procedure is performed (e.g., relative to a robot, navigational guidance system, etc.) Any image-guided surgical procedure which spatially relates pre-operative data to intra-operative execution requires solution of the registration problem.

There are many approaches to registration for image-guided surgery and an excellent review can be found in [5]. A class of registration methods referred to as *shape-based* methods uses representations of object shape to estimate the required transformation. Representations are constructed using data collected in the two coordinate systems (i.e., pre- and intra-operative). Registration estimates a transformation which aligns one shape representation with the other in a manner which minimizes a measure of the distance between them.

Several factors affect shape-based registration accuracy, including: errors in the shape representations due to sensor noise or shape reconstruction errors [10]; the quantity of registration data; and the locations on the registration object from which the data are collected [11]. This paper addresses the problem of improving shape-based registration accuracy via intelligent selection of registration data and online estimation of accuracy. Intelligent data selection (IDS) is comprised of *geometric constraint analysis* which provides a sensitivity measure shown to be well correlated with registration accuracy; and *geometric constraint synthesis*, an optimization process which generates data configurations which maximize the sensitivity measure for a fixed quantity of data. IDS uses the pre-operative shape representation to generate a *data collection plan* (DCP) which can be used during surgery to guide the acquisition of registration data. *Online accuracy estimation* provides an upper bound on true registration accuracy based upon a conventional root-mean-squared error.

The proposed methods have been investigated in-vitro on cadaveric specimens and via simulation studies and are currently being incorporated into a clinical image-guided orthopaedic surgical application [9]. The current paper describes the methods, reports encouraging results, and suggests approaches for incorporating the methods into clinically viable registration systems.

2. Methods

This paper focuses on a special case of shape-based registration: surface-based registration with discrete point data. One shape representation (the "Model") is a triangle mesh surface model of the registration object constructed from CT images. The other representation (the "Data") is a set of discrete point data collected from the registration object during surgery using a digitizing probe.

2.1. Constraint Analysis

Most approaches to shape-based registration attempt to minimize an error measure such as the following least-squared measure:

$$\min_{T} \sum_{i} \|M_i - T(D_i)\|^2 \qquad (1)$$

where each D_i represents a point in the Data, each M_i represent a point in the Model, and T is a 3-D transformation which minimizes the expression. Details of shape-based registration methods can be found in [2][3][5], and descriptions of the methods used in this work appear in [8]. Due to fundamental similarities among shape-based registration solution methods, the techniques proposed in this paper are independent of the particular registration solution method used.

Solving the registration problem results in an estimate, T_{est}, of the true (and usually unknown) registration transformation, T_{true}. The error resulting from a single registration trial can be expressed as:

$$T_{err} = T_{true} \cdot T_{est}^{-1} \qquad (2)$$

where T_{err} is a transformation which represents the difference between estimated and true transformations. The goal of constraint analysis is to provide a scalar measure of sensitivity which is a good predictor of T_{err} for a given Model and Data *without performing registration, and without the need to know* T_{true}.

2.1.1. Derivation of the Method

The point-to-surface distance in (1) is defined as the length of the shortest line joining a point and a surface. In general, there is no closed form analytical expression for this distance given an arbitrary surface; however, the following local approximation has been proposed [12]:

$$D(x) = \frac{F(x)}{\|\nabla F(x)\|} \qquad (3)$$

where x is a point which may or may not lie on the surface, $F(x) = 0$ is the implicit equation of the surface, $\|\nabla F(x)\|$ is the magnitude of the gradient of F at x, and $D(x)$ is the approximate distance. It can be shown that $D(x)$ is a first order approximation of the true point-to-surface distance, and is exact when the surface is a plane.

Given a point x_s which lies on the surface, a small transformation, T_s, will perturb this point from its resting position. T_s can be represented by a homogeneous transformation which is a function of the 6 parameter vector,

$$t = \begin{bmatrix} t_x & t_y & t_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathrm{T}} \qquad (4)$$

in which $(\omega_x, \omega_y, \omega_z)$ are rotations about the X, Y, and Z axes respectively, and (t_x, t_y, t_z) are translations along the newly rotated X, Y, and Z axes. The gradient of D with respect to t is a 6-vector,

$$V(x_s) = \frac{\partial}{\partial t} D(T_s(x_s)) = \begin{bmatrix} n \\ x_s \times n \end{bmatrix} \qquad (5)$$

where n is the unit normal to the surface at the point x_s [8]. This result can be extended to consider the effect of perturbing a *collection* of points with respect to the surface:

$$E_P(T_s(x_s)) = dt^{\mathrm{T}} \left[\sum_{x_s \in P} V(x_s) V^{\mathrm{T}}(x_s) \right] dt \qquad (6)$$

$$= dt^{\mathrm{T}} \Psi_P dt$$

The scalar quantity $E_P(T_s(x_s))$ is a first order approximation of the least-squared error of (1). It measures the error which would result by perturbing a set of discrete points, P, initially assumed to be on the surface, by the small transformation T_s. The matrix Ψ_P is a symmetric, positive semi-definite 6x6 scatter matrix which contains information about the distribution of the original $V(x_s)$ vectors over the points in the set P. Performing principal component analysis [4], Ψ_P is transformed into an expression which is more easily interpreted:

$$E_P(\boldsymbol{T}_s(\boldsymbol{x}_s)) = dt^{\mathrm{T}} Q \Lambda Q^{\mathrm{T}} dt = \sum_{i=1}^{6} \lambda_i (dt^{\mathrm{T}} \boldsymbol{q}_i)^2 \qquad (7)$$

where $\Lambda = diag[\lambda_1 \ldots \lambda_6]$ is a diagonal 6x6 matrix of the eigenvalues of Ψ_P in which $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \lambda_5 \geq \lambda_6$; Q is a 6x6 matrix whose columns are the eigenvectors of Ψ_P; and each $\boldsymbol{q}_i$ is an eigenvector corresponding to the eigenvalue λ_i which represents a differential transformation 6-vector. This result is similar to one presented in the context of industrial inspection [6].

From (7) it can be seen that the eigenvector $\boldsymbol{q}_1$ corresponding to the largest eigenvalue, represents the *direction of maximum constraint.* Perturbing the points in the set P in the direction of $\boldsymbol{q}_1$ will result in the largest possible change in E_P from among all possible directions of perturbation. Similarly, the differential transformation represented by the eigenvector $\boldsymbol{q}_6$ corresponds to the *direction of maximum freedom.* Perturbing the points in this direction will result in the smallest possible change in E_P from among all possible directions of perturbation. In general, an eigenvalue λ_i is proportional to the rate of change of E_P induced by a differential transformation in the direction $\boldsymbol{q}_i$.

A special situation occurs when some of the λ_i are close to or equal to zero. For each such eigenvalue, a singularity exists such that perturbing the points in the direction of the corresponding eigenvector will result in no change in E_P. Such singularities are undesirable in registration since it is impossible to localize the object in the direction(s) of the singularity(s). As demonstrated below, sets of discrete points, P, which have well-conditioned scatter matrices, Ψ_P, are preferable to sets which have ill-conditioned scatter matrices for achieving accurate registration. In this work, the noise amplification index (NAI) [7] is used as a measure of matrix conditioning and is defined as

$$\frac{\lambda_6}{\sqrt{\lambda_1}} \qquad (8)$$

This quantity is the product of the inverse condition number and the square root of the minimum eigenvalue, and provides an upper bound on the amplification of residual errors (e.g., discrete point Data measurement noise, and errors in the Model) to the estimated parameters (e.g., registration transformation parameters) [7].

2.1.2. Scale and Coordinate System Dependences

In constraint analysis, there is an implicit weighting factor related to object size which determines the relative importance of rotational versus translational errors. Due to the $\boldsymbol{x}_s$ term on the right hand side of (5), if constraint analysis is applied to two objects which differ only in size, the resulting NAI values will differ. The larger object will weight rotational components more heavily since the corresponding $\boldsymbol{x}_s$ terms will be larger. A solution to this problem is to pre-normalize the Model so that the average radius as measured about the origin is unity. This has the effect of weighting rotational and translational components equally, on average. A complete discussion of the scale dependence problem can be found in [8].

Constraint analysis has a dependence upon the location of the origin of the Model coordinate system arising from the $\boldsymbol{x}_s$ term on the right hand side of (5). For a given Model, it can be shown that maximal sensitivity of constraint analysis is achieved when the constraint analysis coordinate system origin is located at the centroid of the Model [8].

2.2. Constraint Synthesis

The goal of constraint synthesis is to automatically generate Data sets which maximize the NAI for a given Model and a fixed number of points. The resulting *data collection plan* (DCP) can then be used to guide the acquisition of Data during the intra-operative Data collection process. More formally, the constraint synthesis problem is to:

Select M discrete points from a set, V, and place them into the set, P, of (6) such that the NAI is maximized.

In this paper, the set V is composed of all vertices of a given triangle mesh Model. In general, any sufficiently dense sampling of a surface can be used for V. If there are regions of the Model in which Data cannot be collected (e.g., because of limited access during surgery), points in these regions can be excluded from V. The number of points, M, in P is fixed for a given trial of constraint synthesis. Finding Data configurations of *minimum size* which satisfy registration accuracy requirements is discussed below.

Constraint synthesis is a combinatorial search problem, and for all but artificially small problems the solution space is too large to search exhaustively. A search algorithm for solving constraint synthesis which combines hillclimbing and a nondeterministic optimization method is described below. A complete description of constraint synthesis solution methods can be found in [8].

2.2.1. Hybrid PBIL / Hillclimbing Search Algorithm

In *next ascent hillclimbing* (NAH), M vertices are chosen from the set of possible vertices, V, and placed into the "selected" set P. Let NAI(P) represent the value of the NAI computed from (5) - (8) using the points in P. Randomly select a vertex, $\mathbf{v}_p$, from P, and a vertex, $\mathbf{v}_v$, from V. Substitute $\mathbf{v}_p$ with $\mathbf{v}_v$ in P, and compute the new value of NAI(P). If this substitution results in an improvement in the NAI, then implement the substitution and iterate the process. If the substitution does not improve the NAI, then recompute NAI(P) with new randomly selected vertices $\mathbf{v}_p$ and $\mathbf{v}_v$. Continue iterating until there are no additional substitutions which improve the NAI. The maximum number of NAI evaluations per iteration is N(M-1) where N and M are the number of vertices in the sets V and P respectively, although such a large number of evaluations is rarely reached in practice. For an average size problem (e.g., $\mathrm{N} \approx 5000, \mathrm{M} \approx 50$), NAH usually converges within 1000 iterations. The number of NAI evaluations is typically small during initial iterations, and increases during the later iterations when there are fewer possible substitutions which increase NAI(P).

In high-dimensionality optimization problems, hillclimbing methods such as NAH are susceptible to local minima in the search space. Genetic algorithms (GAs) are biologically motivated adaptive systems based upon principles of natural selection and genetic recombination which attempt to avoid local minima. A simplified model of the GA called Population-Based Incremental Learning (PBIL) was recently introduced [1]. For the purposes of this paper, PBIL can be thought of as a black-box with the following inputs: the set of allowable vertices, V; the number of points in the configuration set, P; a function which computes NAI(P) based upon the surface Model; and a stopping criterion. The output of PBIL is the particular configuration which maximizes the NAI among all configurations evaluated by PBIL within a given trial.

While PBIL is good at avoiding local minima in the constraint synthesis search space, the resulting solutions may not be locally optimal. Likewise, hillclimbing methods are good at ensuring local optimality, but usually don't converge to globally optimal configurations. By combining these two approaches, it is possible to take advantage of the strengths of each. In the hybrid search algorithm, PBIL is run, followed by a run of NAH initialized at the configuration found by PBIL.

2.3. Data Configuration Stability

Data collection plans (DCPs) generated by constraint synthesis can be used to guide acquisition of registration data. Since the precise object location is unknown before registration, it is impossible to acquire the exact points specified by constraint synthesis. Due to this uncertainty and to noise in the sensing process, the *effective* NAI value (i.e., computed from the collected Data) may be smaller than the *ideal* NAI (i.e., computed from the DCP). Certain Data configurations are more stable than others (i.e., there is less NAI variation as the points in P are perturbed about the DCP positions). Attempts to incorporate stability criteria into the constraint synthesis process have resulted in exponential complexity [8]. Nevertheless, improved stability can be achieved via two methods: navigational guidance during Data collection and high curvature filtering.

During the data acquisition process, it is possible to use the current registration transformation estimate to provide navigational guidance to the human

Data collector. Guidance is provided by displaying a 3-D graphical rendering of the registration object and overlaying icons representing the locations of the desired point and the sensor tip. The sensor location icon is dynamically updated in real-time based upon measurements, and is derived from the registration transformation estimate. The goal of the Data collector is to align the two icons. Each time additional Data is collected, uncertainty in the collection process is reduced by refining the registration transformation estimate.

The primary cause of Data configuration instability is disparity between the surface normals of desired and collected Data points. Constraint synthesis may select a given Data point because its surface normal strongly contributes to constraint in a given direction (see (5)). However, if the Data point actually collected has a significantly different surface normal, the resulting NAI value may be reduced. This effect can be reduced by initially focussing data collection in regions of low curvature so that surface normals of the collected points are more likely to be similar to those of the desired points. After low curvature points are collected and collection uncertainty is reduced, points in higher curvature regions can be collected. To implement this, several DCPs are synthesized, some with points in regions of low curvature and others in regions of higher curvature [8]. The resulting DCPs can then be used to guide the collection process.

3. Results

This section demonstrates significant improvement in registration accuracy due to the proposed methods. Three Models are used in the reported experiments: a cube with edge length of 100 mm, a human cadaveric femur, and a human cadaveric pelvis. Models of the femur and pelvis with superimposed Data collection plans are shown in Figure 1.

The registration error measure used to report results in this section is the maximum correspondence error (MCE) [8][11]. The MCE is computed by transforming all vertices in a Model by $\boldsymbol{T}_{err}$ of (2), computing distances between each transformed vertex and its un-transformed correspondence, and selecting the largest distance. The MCE specifies the largest single point displacement within a registration object resulting from $\boldsymbol{T}_{err}$

3.1. Constraint Analysis Experiments

Registration trials were conducted using simulated Data to demonstrate the relation between registration error and NAI. Data points were generated by applying known random transformations to nominal Data configurations and adding zero-mean Gaussian noise. Since the true transformations, $\boldsymbol{T}_{true}$, are known, the error transformations, $\boldsymbol{T}_{err}$, can be computed. Figure 2 shows a plot of MCE vs. NAI for the three nominal cube configurations shown on the right of the figure. Configuration C1 contains 24 points per face, while C2 and C3 contain 4 points per face each. For each configuration, the mean, standard deviation, minimum and maximum MCE over 500 registration trials are plotted. The parameters for generating noise and random transformations are shown in the plot. The trend from the plot is clear: configurations with larger values of NAI result in smaller registration error. In particular, note that configuration C2 has smaller values of MCE (and a larger NAI) than C3, despite having the same number of points.

Figure 3 demonstrates differences in noise sensitivity as a function of NAI for the cube configurations. The graphs show how MCE varies as a

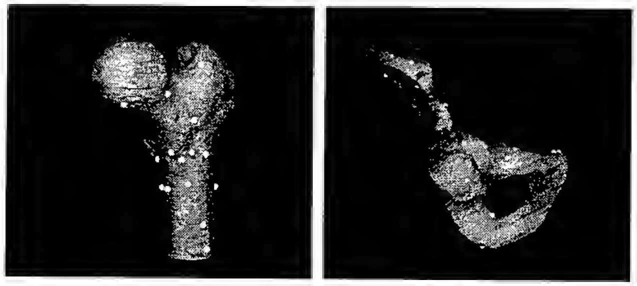


Figure 1: Surface Models of the femur and pelvis with overlaid DCPs.

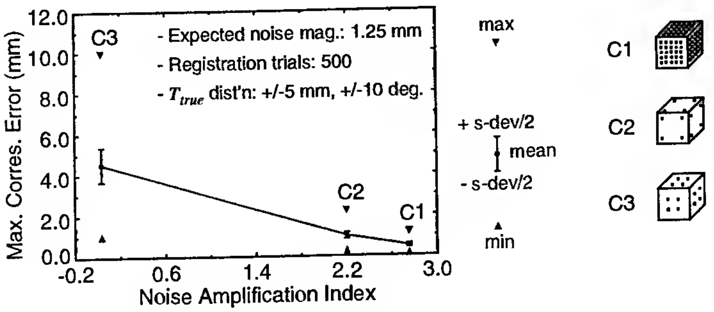


Figure 2: MCE vs. NAI for 3 configurations on a cube.

TABLE 1. Pelvis synthesis results - NAI values (max / min).

Method	Configuration Size - M			
	10	25	50	75
Random	0.41 / 0.00	1.18 / 0.08	1.52 / 0.33	1.68 / 0.53
NAH	1.42 / 1.28	2.62 / 2.43	3.97 / 3.76	4.90 / 4.79
PBIL	1.41 / 1.35	2.70 / 2.58	3.88 / 3.81	4.84 / 4.76
PBIL + NAH	1.52 / 1.36	2.75 / 2.65	4.02 / 3.92	4.94 / 4.90

function of expected noise magnitude. For each datum, 500 registration trials were performed and the mean values for these trials are plotted. In the absence of noise, all three configurations perform equally well. As noise increases, configurations with smaller values of NAI are clearly more sensitive. This illustrates that the utility of intelligent data selection is dependent upon the magnitude of sensor noise (among other factors).

3.2. Constraint Synthesis Experiments

Table 1 demonstrates the efficacy of the constraint synthesis search algorithms for the pelvis. Data configurations were synthesized using 4 configuration sizes and 4 methods of generation. Five configurations were generated for each size-method combination, except for the random method for which 1000 configurations were generated. The maximum and minimum NAI values over the generated configurations are shown. For each configuration size, the hybrid PBIL/NAH method produced the best results.

Figure 4 compares 5 random and 5 synthesized configurations of size 25 for the pelvis in a plot of MCE versus NAI. For each configuration, a set of registration trials was performed using the indicated parameters. In this graph, the 5th and 95th percentiles of MCE are plotted instead of the minimum and maximum values. When generating the simulated registration Data, a second noise component was added which models the uncertainty associated with Data collection. This noise perturbs a

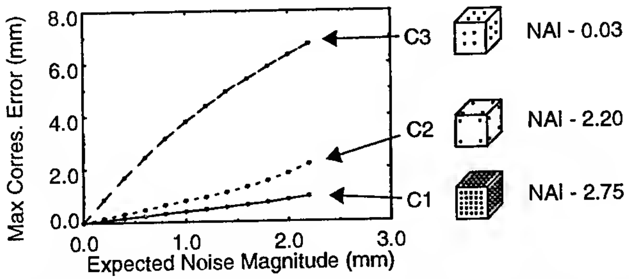


Figure 3: MCE vs. expected noise magnitude for 3 cube configurations.

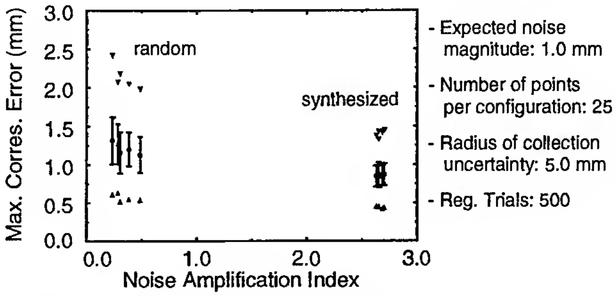


Figure 4: MCE vs. NAI for 5 random and 5 synthesized configurations on pelvis Model.

point from its nominal location by a uniform random distance *along the surface*. For this experiment, the radius of uncertainty was 5.0 mm. From the graph, it is clear that the synthesized configurations are superior to the randomly generated ones in terms of both NAI and MCE.

Figure 5 shows similar results for the femur Model using 5 random and 5 synthesized configurations of size 10. The figure demonstrates the effect of high curvature filtering; no filtering results in unstable Data configurations and larger errors.

3.3. In-vitro Cadaver Experiments

We performed registration trials using Data collected from a cadaveric femur. For these experiments, estimation of T_{true} is a challenging engineering problem which our group has solved using a highly accurate fiducial-based registration method [11]. Using a filtered version of the femur Model, DCPs of 6 and 50 points were synthesized, each a total of 5 times. The corresponding Data points were collected on the actual femur using a digitizing probe. Each synthesized configuration was independently collected 5 times. In addition, 50 manually-selected Data sets were collected for each configuration size. To guide the collection process, the navigational guidance mechanism described above was used. Initial values of T_{est} were computed using manually selected anatomical landmarks and point-to-point registration [2].

Experimental results are shown in Figure 6. Each graph plots the MCE value resulting from registration versus the *effective* NAI value computed after registration using the closest Model points $(M_i$ of (1)) to solve for $\boldsymbol{n}$ and $\boldsymbol{x}_s$ of (5). From the graphs it is clear that the synthesized point configurations are superior to the manually selected ones for both configuration sizes. Six points is the theoretical minimum number required to solve the shape-based registration problem without correspondence. As seen, selecting 6 well-conditioned Data points is a difficult task for humans. Note that some synthesized configurations for the 6-point results have small NAI values and large MCE values due to data collection uncertainty. However, using the online accuracy evaluation method described below, such configurations can easily be identified and additional Data can be collected to improve the result.

To be useful, an online accuracy estimate must relate a quantity which can be measured during the registration process, to a second quantity which has

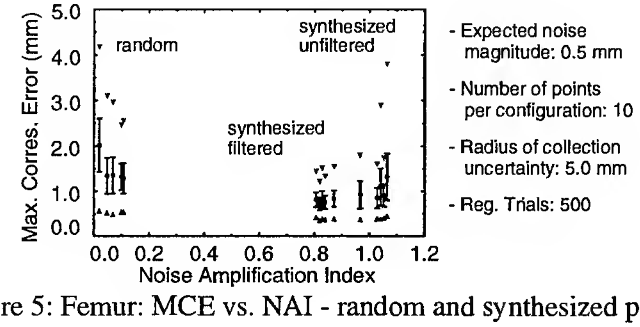


Figure 5: Femur: MCE vs. NAI - random and synthesized points.

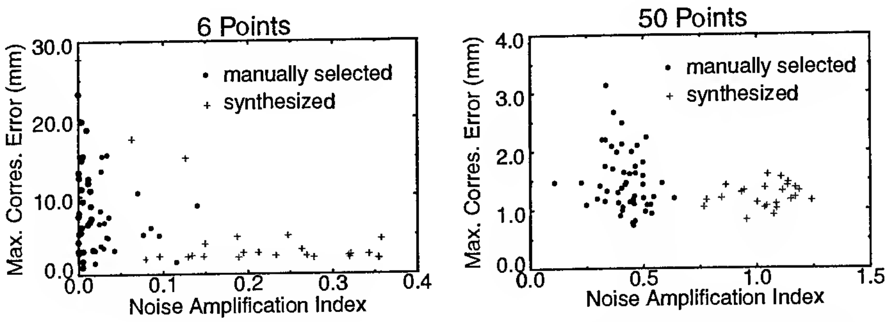


Figure 6: MCE vs. NAI - physically collected Data on femur. Note scale differences.

physical meaning to the task for which registration is being performed. Figure 7 shows a plot of MCE versus RMS error (definition in the figure). It is shown in [8] that the slope of the line which relates worst case MCE to RMS error is independent of sensor noise, the number of Data points, and Data collection uncertainty, assuming that the effective NAI value is slightly greater than zero. Furthermore, it is shown that the slope of this line can be determined from simulated registration experiments such as those of Section 3.2.. Therefore, during the registration process, online measurement of RMS error can be used to estimate an upper bound on MCE. This estimate can then be used to determine if accuracy requirements are satisfied, and additional Data collection can be requested if not.

By coupling online accuracy estimation with intelligent data selection, it is possible to collect *minimally-sized* Data sets which satisfy accuracy requirements. This is done by pre-synthesizing multiple NAI-optimal configurations of increasing size, each of which is a superset of the previous. During the collection process, a Data set is collected and registered, and accuracy is estimated. This process is continued until accuracy requirements are satisfied, or until all of the synthesized sets have been collected.

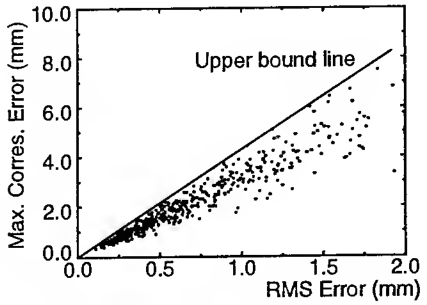
4. Conclusions

The methods described in this paper show promise as tools for analyzing and maximizing accuracy in shape-based registration. Intelligent data selection is likely to be most useful when data collection is expensive and sensor noise is high. Online accuracy estimation is likely to be useful with and without intelligent data selection. Work is currently in progress to evaluate the practicality of these methods in clinical situations.

References

[1] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In A. Prieditis, ed, *Proc. Int'l Conf. Mach. Learning*, pp. 38–46, San Mateo, CA, 1995. Morgan Kaufmann Publishers.

[2] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE. PAMI*, 14(2):239–256, Feb 1992.



$$RMS = \sqrt{\sum_i \|M_i - T(D_i)\|^2}$$

Figure 7: Basis of online accuracy estimation - plot of MCE vs. RMS error.

[3] E. Cuchet et al., Registration in neurosurgery and neuroradiotherapy applications. In *Proc 2nd Int'l Symp. MRCAS*, pp. 31–38, Baltimore, Nov. 1995.

[4] M. G. Kendall and A. Stuart. *Canonical Variables*, chap 43, pp. 320–369. Griffin, London, 4th ed., 1977.

[5] S. Lavallee. Registration for computer-integrated surgery: Methodology, state of the art. In R. H. Taylor, et al., eds, *Computer-Integrated Surgery*, chap 5, pp 77–97. MIT Press, Cambridge, Massachusetts, 1995.

[6] C.H. Menq, H.T. Yau, and G.Y. Lai. Automated precision measurement of surface profile in CAD-directed inspection. *IEEE Trans. Robotics and Automation*, 8(2):268–278, April 1992.

[7] A. Nahvi and J.M. Hollerbach. The noise amplification index for optimal pose selection in robot calibration. In *Proc IEEE Int'l Conf. Robotics and Automation*, Minneapolis, April 1996.

[8] D. A Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, December 1996.

[9] D. A. Simon, et al., Development and validation of a navigational guidance system for acetabular implant placement. In *Proc. 1st Joint CVRMed / MRCAS Conference*, Grenoble, March 1997.

[10] D. A. Simon, et al., Accuracy validation in image-guided orthopaedic surgery. In *Proc. 2nd Int'l Symp. MRCAS*, Baltimore, Nov. 1995.

[11] D. A. Simon, M. Hebert, and T. Kanade. Techniques for fast and accurate intra-surgical registration. *Journal of Image Guided Surgery*, 1(1):17–29, April 1995.

[12] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans PAMI*, 13(11):1115–1138, Nov 1991.

Acknowledgments

This work was supported in part by a National Challenge grant from the NSF (IRI-9422734).

Consensus Surfaces for Modeling 3D Objects from Multiple Range Images

Mark D. Wheeler
Apple Computer, Inc.
1 Infinite Loop, MS: 301-3M
Cupertino, CA 95014
email: mdwheel@apple.com

Yoichi Sato
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
email: ysato@cs.cmu.edu

Katsushi Ikeuchi *
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
email: ki@cs.cmu.edu

Abstract

In this paper, we present a robust method for creating a triangulated surface mesh from multiple range images. Our method merges a set of range images into a volumetric implicit-surface representation which is converted to a surface mesh using a variant of the marching-cubes algorithm. Unlike previous techniques based on implicit-surface representations, our method estimates the signed distance to the object surface by finding a consensus of locally coherent observations of the surface. We call this method the consensus-surface algorithm. This algorithm effectively eliminates many of the troublesome effects of noise and extraneous surface observations without sacrificing the accuracy of the resulting surface. We utilize octrees to represent volumétric implicit surfaces—effectively reducing the computation and memory requirements of the volumetric representation without sacrificing resolution (and, hence, accuracy) of the volume grid. Our results demonstrate that our consensus-surface algorithm can construct accurate geometric models from rather noisy input range data and somewhat imperfect alignment.

1 Introduction

In this paper, we present a novel approach for building a 3D surface model from a number of range images of an object. The goal of this work is to use real images of an object to automatically create a model which is:

- Geometrically accurate: depicts the correct dimensions of the object and captures small details of the object geometry
- Clean: eliminates noise and errors in the views
- Complete: models the surface as much as is observable from the sample views

We begin by reviewing three methods which are most closely related to our work and follow that by a brief discussion of other related work. The first three works are similar to our algorithm in that they all make use of volumetric, implicit-surface representation and the marching-cubes algorithm [Lorensen and Cline, 1987] to merge the range-image data from several views into a surface model. The main differences between these algorithms are their methods for computing the signed distance from each voxel (volume element) to the closest surface.

Hoppe et al. [Hoppe *et al.*, 1992] were the first to propose constructing 3D surface models by applying the marching-cubes algorithm [Lorensen and Cline, 1987] to a discrete, implicit-surface function generated from a set of range images. Their algorithm computes the signed distance function from a set of 3D points. Much of their algorithm works to infer local surface approximations from the cloud of points. Nearest-neighbor search of the inferred surfaces is used to compute the signed distance from each voxel to the surface of the point set.

Curless and Levoy [Curless and Levoy, 1996] followed Hoppe's general scheme with a few significant departures. First, their method was geared towards using 3D data acquired from range images. Rather than performing a simple search for the closest point from a voxel's center to determine the signed distance, They take a weighted average of the signed dis-

*This research has been supported in part by the Advanced Research Projects Agency under the Department of the Army, Army Research Office grant number DAAH04-94-G-0006, and in part by the Office of Naval Research grant number N00014-93-1-1220. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army, the Department of the Navy or the United State Government.

tances from the voxel center to range-image points whose image rays intersect the voxel—integrating the signed distance along these rays through the volume.

The method most similar to our work is that of Hilton et al. [Hilton *et al.*, 1996]. As in our work and Curless and Levoy's work, Hilton et al. generate a volumetric implicit-surface representation from a number of triangle sets obtained from range images. Similarly to our algorithm, Hilton's method uses a nearest-neighbor search to find the surface points from each view which are closest to a given voxel's center. Heuristics are used to determine which closest points to use in computing the signed distance.

The main limitation of the above algorithms is that they do not compensate for noise or extraneous point data—the data is assumed to be part of the object and noise is assumed to be negligible. In addition, each of these methods can suffer from inaccuracy due to their integration strategy (for more details see [Wheeler, 1996]).

There have been several other approaches to this modeling problem; most notably, Soucy and Laurendre [Soucy and Laurendeau, 1992] and Turk and Levoy [Turk and Levoy, 1994] presented methods for piecing together sets of triangulated surfaces. Both methods marked significant advancements in the state of the art, but often perform poorly with respect to noise and alignment errors in the data.

2 Approach

The problem we are tackling in this paper is to build a 3D model, a unified surface representation, from a number of range images of an object. To build a 3D surface model from multiple range images, we must address the following problems:

- View alignment: To merge the data, the data from all the images must be in the same coordinate system.
- Data merging: We need to merge all the image data while eliminating or greatly reducing the effects of noise and extraneous data.

Our solution makes use of a volumetric representation to avoid difficulties associated with topology. We will show how the volumetric representation simplifies our data-merging problem—virtually eliminating the topology issue. The volumetric representation can be conveniently converted into a triangulated mesh representation with little loss of geometric accuracy. The merging problem is then a matter of converting our input surface data to the volumetric representation.

The conversion problem is exacerbated by the fact that input surface data from real sensors (e.g., range sensors or stereo) is noisy and, in fact, will contain surfaces that are not part of the object we are interested in modeling. Our method for merging the surfaces into a volumetric representation takes full consideration of these facts to best take advantage of the multiple observations to smooth out the noise and eliminate undesired surfaces from the final model.

One important issue which we do not address in this work is how to select views in order to best cover the surface. The human operator determines the number of views and the object orientation for each view.

The rest of this paper provides the details of our solutions to these problems which combine to form a practical method for building 3D surface models from range images of an object.

3 View Alignment

After taking several range images of an object and converting them to surfaces (described in [Wheeler, 1996]), we need to eventually merge all these surfaces into a single model. The problem is that each view is taken from a different coordinate system with respect to the object. In order to compare or match the data from different views, we must transform all the data into the same coordinate system with respect to the object.

There are several ways we can approach this alignment problem—each requiring varying levels of human interaction: manual alignment, semi-automatic alignment, and automatic alignment. As view alignment is not the central focus of this work, we use controlled motion with calibration as it is currently the most practical option for an automatic solution. In our experimental setup, we calibrate two axes of a Unimation Puma robot with respect to a range-sensor coordinate frame. We can then mount the object on the robot's end effector and acquire images of an object at arbitrary orientations with known positions.

From this point we assume that the views are aligned. Next, we consider the problem of merging all the data from these views into a single model of the object's surface.

4 Data Merging

Given a number of triangle sets which are aligned with respect to the desired coordinate system, we are now faced with the task of taking many triangulated surfaces in 3D space and converting them to a triangle patch surface model. Even if we are given perfect sets of triangulated surfaces from each view which are more or less perfectly aligned, the merging problem is difficult. The problem is that it is difficult to determine how to connect triangles from different surfaces without knowing the actual surface beforehand. There are an exponential number of ways to connect two triangulated surfaces together, some acceptable and some not acceptable. This problem is exacerbated by noise in the data and errors in the alignment. Not only does the determination of connectedness become more difficult, but now the algorithm must also consider how to eliminate the noise and small alignment errors from the resulting model. Recently, however, several researchers [Hoppe *et al.*, 1992, Curless and Levoy, 1996, Hilton *et al.*, 1996] have moved from trying to connect together surface patches from different views to using volumetric methods which hide the topological problems—making the surface-merging problem more tractable. In the next section we discuss the volumetric method which we use to solve the surface-merging problem.

4.1 Volumetric Modeling and Marching Cubes

Recently, the marching-cubes algorithm [Lorensen and Cline, 1987], an algorithm developed for graphics modeling applications, has made volumetric modeling more useful by virtually eliminating the blocky nature of occupancy grids. The representation used by the marching-cubes algorithm is slightly more complicated than the occupancy-grid representation. Instead of storing a binary value in each voxel to indicate if the cube is empty or filled, the marching-cubes algorithm requires that the data in the volume grid are samples of an implicit surface. In each voxel, we store the signed distance, $f(\mathbf{x})$, from the center point of the voxel, $\mathbf{x}$, to the closest point on the object's surface. The sign indicates whether the point is outside, $f(\mathbf{x}) > 0$, or inside, $f(\mathbf{x}) < 0$, the object's surface, while $f(\mathbf{x}) = 0$ indicates that $\mathbf{x}$ lies on the surface of the object.

The marching-cubes algorithm constructs a surface mesh by "marching" around the cubes while following the zero crossings of the implicit surface $f(\mathbf{x}) = 0$. The location of the surface can be interpolated by examining the signed distances of neighboring voxels. Thus, the resulting surface will be relatively smooth and the accuracy of the surface will be greater than the resolution of the volume grid.

The marching-cubes algorithm and the volumetric implicit-surface representation provide an attractive alternative to other conceivable mesh-merging schemes. First, they eliminate the global topology problem—how the various surfaces are connected—for merging views. The representation can model objects of arbitrary topology as long as the grid sampling is fine enough to capture the topology. Most importantly, the problem of creating the volumetric representation can be reduced to a single, simple question:

> *What is the signed distance between a given point and the surface?*

The given point is typically the center of a voxel, but we don't really care. If we can answer the question for an arbitrary point, then we can use that same question at each voxel in the volume.

Now we may focus on a more easily defined problem: How do we compute $f(\mathbf{x})$? The real problem underlying our simple question is that we do not have a surface; we have many surfaces, and some elements of those surfaces do not belong to the object of interest but rather are artifacts of the image acquisition process or background surfaces. In the next subsection, we present an algorithm that answers the question and does so reliably in spite of the existence of noisy and extraneous surfaces in our data.

4.2 Consensus-Surface Algorithm

In this section, we will answer the question of how to compute the signed distance function $f(\mathbf{x})$ for arbitrary points $\mathbf{x}$ when given N triangulated surface patches from various views of the object surface. We call our algorithm the *consensus-surface algorithm.*

We can break down the computation of $f(\mathbf{x})$ into two steps:

- Compute the magnitude: compute the distance, $\mid f(\mathbf{x}) \mid$, to the nearest object surface from $\mathbf{x}$
- Compute the sign: determine whether the point is inside or outside of the object

We are given N triangle sets—one set for each range image of our object—which are aligned in the same coordinate system. The triangle sets are denoted by T_i, where $i = 1, ..., N$, The union of all triangle sets is denoted by $T = \bigcup_i T_i$. Each triangle set, T_i, consists of some number n_i of triangles which are denoted by $\tau_{i,j}$, where $j = 1, ..., n_i$.

If the input data were perfect (i.e., free of any noise or alignment errors in the triangle sets from each view), then we could apply the following *naive algorithm*, Algorithm *ClosestSignedDistance*, to compute $f(\mathbf{x})$:

```
**Algorithm** *ClosestSignedDistance*
**Input:** point $\mathbf{x}$
**Input:** triangle set $T$
**Output:** the signed distance $d$
$\langle \mathbf{p}, \hat{\mathbf{n}} \rangle \leftarrow$ *ClosestSurface*$(\mathbf{x}, T)$
$d \leftarrow \| \mathbf{x} - \mathbf{p} \|$
**if** $(\hat{\mathbf{n}} \cdot (\mathbf{x} - \mathbf{p}) < 0)$
    **then** $d \leftarrow -d$
**return** $d$
```

where Algorithm *ClosestSurface* returns the point, $\mathbf{p}$, and its normal, $\hat{\mathbf{n}}$, such that $\mathbf{p}$ is the closest point to $\mathbf{x}$ from all points on triangles in the triangle set T.

The naive algorithm for $f(\mathbf{x})$ finds the nearest triangle from all views and uses the distance to that triangle as the magnitude of $f(\mathbf{x})$. The normal of the triangle can be used to determine whether $\mathbf{x}$ is inside or outside the surface. If the normal of the closest surface point is directed towards $\mathbf{x}$, then $\mathbf{x}$ must be outside the object surface.

Again, the naive algorithm will work for perfect data. However, we must consider what happens when we try this idea on real data. The first artifact of real sensing and small alignment errors is that we no longer have a single surface, but several noisy samples of a surface (see Figure 4 in the results section for an example of the type of noise that may be present). We are now faced with choices on how to proceed. Clearly, choosing the nearest triangle (as in Algorithm *ClosestSignedDistance*) will give a result as noisy as the constituent surface data. For example, a single noisy bump from one view can result in a bump on the final model.

Inconsistent values for the implicit distances will appear when a voxel center is on or near a surface, since the samples will be randomly scattered about the real surface location. It may become difficult to differentiate between several observations of a single surface or a thin wall. This is especially a problem if the noise is of similar scale to the voxel size.

A more sinister problem for the naive algorithm applied to real images is the existence of noise and extraneous data. For example, it is not uncommon to see triangles sticking out of a surface or other triangles that do not belong to the object. This can occur due to sensor noise, quantization, specularities and other possibly systematic problems of range imaging. This makes it very easy to infer the incorrect distance and more critically the incorrect sign, which will result in very undesirable artifacts in the final surface. One badly oriented triangle can result in a voxel which is assigned $f(\mathbf{x})$ with the incorrect sign. The result will be in a hole rising out of the surface produced by the marching-cubes algorithm.

Our solution to these problems is to estimate the surface locally by averaging the observations of the *same surface*. The trick is to specify a method for identifying and collecting all observations of the same surface.

Nearby observations are compared using their location and surface normal. If the location and normal are within a predefined error tolerance (determined empirically), we can consider them to be observations of the same surface. Given a point on one of the observed triangle surfaces, we can search that region of 3D space for other nearby observations from other views which are potentially observations of the same surface. This search for nearby observations can be done efficiently using k-d trees [Friedman *et al.*, 1977].

If an insufficient number of observations of a given surface are found, then the observation can be discarded as isolated/untrusted and the search can continue. Thus, we are requiring a quorum of observations before using them to build our model. The quorum of observations can then be averaged to produce a *consensus surface*. This process virtually eliminates the problems described previously (with respect to the naive algorithm).

As an improvement over using an equally weighted voting scheme, we can assign a confidence value ω to each input surface triangle: weighting the surface points/triangles from a range image by the cosine of the angle between the viewing direction and the surface normal. This is simply computed by

$$\omega = \hat{\mathbf{v}} \cdot \hat{\mathbf{n}}$$

where $\hat{\mathbf{v}}$ and $\hat{\mathbf{n}}$ are the viewing direction (of the given point in the range image) and normal, respectively, of the given triangle. The consensus can now be measured as a sum of confidence measures and the quorum is over this weighted sum. The rationale is that two low-confidence observations should not have the same impact on the result as two high-confidence observations. We can now specify the consensus-surface algorithm:

```
Algorithm ConsensusSignedDistance
Input: point $\mathbf{x}$
Input: triangle set $T$
Output: the signed distance $d$
1.  $\langle \mathbf{p}, \hat{\mathbf{n}} \rangle \leftarrow ClosestConsensusSurface(\mathbf{x}, T)$
2.  $d \leftarrow \| \mathbf{x} - \mathbf{p} \|$
3.  if $(\hat{\mathbf{n}} \cdot (\mathbf{x} - \mathbf{p}) < 0)$
4.     then $d \leftarrow -d$
5.  return $d$
```

The only change from Algorithm *ClosestSignedDistance* is that Algorithm *ConsensusSignedDistance* computes the closest consensus-surface point and its normal in line 1. The algorithm for computing the closest consensus-surface point and its normal is as follows:

```
Algorithm ClosestConsensusSurface
Input: point $\mathbf{x}$
Input: triangle sets $T_i$, $i = 1..N$
Output: the point and normal vector pair $\langle \mathbf{p}, \hat{\mathbf{n}} \rangle$
1.  $O_{set} \leftarrow \emptyset$
(* $O_{set}$ is the set of non-consensus neighbors *)
2.  $C_{set} \leftarrow \emptyset$
(* $C_{set}$ is the set of consensus neighbors *)
3.  for each triangulated set $T_i$
4.      do $\langle \mathbf{p}, \hat{\mathbf{n}} \rangle \leftarrow ClosestSurface(\mathbf{x}, T_i)$
5.         $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle \leftarrow$
6.         $ConsensusSurface(\mathbf{p}, \hat{\mathbf{n}}, T)$
7.         if $\omega \geq \theta_{quorum}$
8.            then $C_{set} \leftarrow C_{set} \cup \langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle$
9.            else $O_{set} \leftarrow O_{set} \cup \langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle$
10. if $C_{set} \neq \emptyset$
11.    then $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle \leftarrow$
12.          $\arg\min_{\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle \in C_{set}} \| \mathbf{x} - \mathbf{p} \|$
13.    else $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle \leftarrow \arg\max_{\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle \in O_{set}} \omega$
```

```
14. return $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle$
```

Algorithm *ClosestConsensusSurface* examines the closest point in each view and searches for its consensus surface if one exists. After computing the closest consensus surfaces for each view, it chooses the closest of those from the consensus set C_{set}. C_{set} contains those locally averaged surfaces whose observations' confidence values sum to at least θ_{quorum} . Note that two consensus surfaces are not differentiated based on their confidence sum ω but rather on their proximity to $\mathbf{x}$. If none of the consensus surfaces exist, the algorithm selects the average surface which has the highest summed confidence out of set O_{set}.

For completeness, we outline Algorithm *ConsensusSurface* which is required by line 5 of Algorithm *ClosestConsensusSurface*. Algorithm *ConsensusSurface* basically finds all surface observations that are sufficiently similar to the given point and normal. These observations are then averaged to generate a consensus surface for the input surface. This algorithm relies on the predicate

$$\text{SameSurface}(\langle \mathbf{p}_0, \hat{\mathbf{n}}_0 \rangle, \langle \mathbf{p}_1, \hat{\mathbf{n}}_1 \rangle) = \begin{cases} \text{True} & (\| \mathbf{p}_0 - \mathbf{p}_1 \| \leq \delta_d) \wedge (\hat{\mathbf{n}}_0 \cdot \hat{\mathbf{n}}_1 \geq \cos \theta_n) \\ \text{False} & \text{otherwise} \end{cases} \tag{1}$$

which determines whether two surface observations are sufficiently close in terms of location and normal direction, where δ_d is the maximum allowed distance and θ_n is the maximum allowed difference in normal directions. Now we present the pseudo code for Algorithm *ConsensusSurface*:

```
Algorithm ConsensusSurface
Input: point $\mathbf{x}$
Input: normal $\hat{\mathbf{v}}$
Input: triangle set $T = \bigcup_i T_i$
Output: the point, normal vector, and the sum of
    the observations confidences $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle$
$\mathbf{p} \leftarrow \mathbf{n} \leftarrow \omega \leftarrow 0$
for $T_i \subset T$
    do $\langle \mathbf{p}', \hat{\mathbf{n}}', \omega' \rangle \leftarrow ClosestSurface(\mathbf{x}, T_i)$
        if SameSurface$(\langle \mathbf{x}, \hat{\mathbf{v}} \rangle, \langle \mathbf{p}', \hat{\mathbf{n}}' \rangle)$
            then $\mathbf{p} \leftarrow \mathbf{p} + \omega' \mathbf{p}'$
                $\mathbf{n} \leftarrow \mathbf{n} + \omega' \hat{\mathbf{n}}'$
                $\omega \leftarrow \omega + \omega'$
$\mathbf{p} \leftarrow \frac{1}{\omega} \mathbf{p}$
$\hat{\mathbf{n}} \leftarrow \frac{\mathbf{n}}{\|\mathbf{n}\|}$
return $\langle \mathbf{p}, \hat{\mathbf{n}}, \omega \rangle$
```

Note that in Algorithm *ConsensusSurface*, the definition of Algorithm *ClosestSurface* was slightly modified to also return the confidence ω' of the closest surface triangle.

We refer to this algorithm as a whole as the consensus-surface algorithm. The following conditions are assumed:

1. Each part of the surface is covered by a number of observations whose confidences add up to more than θ_{quorum}.
2. No set of false surfaces with a sufficient summed confidence will coincidentally be found to be similar (following the definition of Equation 1) or this occurrence is sufficiently unlikely.
3. Given N surface views, the real surface is closest to $\mathbf{x}$ in at least one view.

If these assumptions are violated, mistakes in the surface mesh will result. From our experiments, a quorum requirement, θ_{quorum}, of 1.5 to 3.0 is usually sufficient to eliminate errors given a reasonable number of views of the object.

4.3 Accuracy and Efficiency

Volumetric modeling involves a tradeoff between accuracy and efficiency. To achieve desired accuracy we must use a dense sampling of the volume. Since the memory requirements of a volume grid is cubic with respect to the density of the sampling for volumetric modeling, the first thing that gets sacrificed is accuracy. With the straightforward use of volume grids, resources are wasted by computing signed distances in parts of the volume that are distant from the surface. For our purposes, the only voxels that we need to examine are those near the surface, a small fraction of the entire volume grid. Curless and Levoy [Curless and Levoy, 1996] alleviate this problem by run length encoding each 2D slice of the volume.

The octree representation [Meagher, 1980] solves both the accuracy and the efficiency problems while keeping the algorithm implementation simple. Instead of iterating over all elements of the voxel grid, we can apply a recursive algorithm on an octree that samples more finely in octants only when necessary—near the surface of the object. The octree in practice reduces the $O(n^3)$ storage and computation requirement to $O(n^2)$ since the surfaces of 3D objects are, in general, 2D manifolds in a 3D space.

4.4 Cost of the Consensus-Surface Algorithm

To get a rough estimate of the cost of our model-building algorithm, let us assume that there are N views being merged and that for each view the triangle set T_i has n triangles. Algorithm *ConsensusSurface* computes the closest surface for each view which on average will be an $O(N \log n)$ operation assuming k-d trees [Friedman *et al.*, 1977] are used. Algorithm *ClosestConsensusSurface* computes the closest surface and then the respective consensus surface for each view, which adds up to a cost of $O(N^2 \log n)$. Assuming that an $M \times M \times M$ voxel grid is used, the modeling algorithm will cost

$O(M^3N^2 \log n)$. However, if octrees are used we may loosely assume that the number of voxels or octree elements which are evaluated will be proportional to the surface area of the object, reducing the average cost to $O(M^2N^2 \log n)$.

5 Experimental Results

Here we present some experimental results of our implementation of the 3D object modeling algorithm described in this paper. For our experiments, we selected a variety of objects to model using our system. We assume that the objects are rigid and opaque. In this paper, we show the results obtained for modeling a toy duck. See [Wheeler, 1996] for a complete description of our experiments.

For the example presented here we used 48 range images, each containing 256 × 240 pixels with each pixel containing a 3D coordinate. The resolution of data is approximately 1 mm, and the accuracy is on the order of 0.5 mm.

Figures 1-2 show the results, including an intensity image of the object, a close-up of some of the triangulated range images used as input to our algorithm (shaded to better indicate the roughness of the original data), a slice of the volume grid where the grey-scale indicates the proximity to a surface point (black closest, white furthest), and three views of the resulting triangulated model. For this example, the input images contained 555,000 triangles and the resulting model contained 27,000. The finest resolution of the voxel grid was 1.8 mm and approximately 4% of the $128 \times 128 \times 128$ volume grid was expanded by the octree. Parameters used were: $\theta_{quorum} = 2.25$, $\delta_d = 3$, and $\theta_n = 45$. Computing time was 52 minutes on an SGI Indy 5 workstation (a 124 MIPS/49.0 MFLOPS machine).

As an example of what the naive algorithm, Algorithm *ClosestSignedDistance* of Section 4.2, would produce we show the example of the result of the naive algorithm on the duck data set in Figure 3. Notice how many extraneous surfaces exist near the duck from the input range-image data. The naive algorithm fails because it trusts that every surface observation is an accurate observation of the object surface. As can be seen from the sample range data of the duck in Figure 1, this is not the case.

To more clearly illustrate the accuracy of our modeling algorithm, Figure 4 shows a cross section of the final model and the original input range-image data. This example demonstrates the ability of our consensus-surface algorithm to accurately locate the surface in very noisy data.

6 Conclusion

We have described a method to create a triangulated surface mesh from N range images. Robotic calibration is used to acquire images of the object under known transformations, allowing us to align the images into one coordinate frame reliably. Our method for data merging takes advantage of the volumetric implicit-surface representation and the marching-cubes algorithm to eliminate topological problems.

The main contribution of this paper is our algorithm for merging data from multiple views: the consensus-surface algorithm which attempts to answer the question

> *What is the closest surface to a given point?*

With the answer to this question, we can easily compute the signed distance $f(\mathbf{x})$ correctly. While other known methods also implicitly address this question, their algorithms do not capture the essence of the problem and produce answers by taking averages of possibly unrelated observations. In contrast, our algorithm attempts to justify the selection of observations used to produce the average by finding a quorum or consensus of locally coherent observations. This process eliminates many of the troublesome effects of noise and extraneous surface observations in our data. Consensus surfaces can be computed independently for any point in the volume. This feature makes it very easy to parallelize and allows us to straightforwardly use the octree representation. The octree representation enables us to model objects with high accuracy with greatly reduced computation and memory requirements.

We have presented the results of our modeling algorithm on a number of example problems. These results demonstrate that our consensus-surface algorithm can construct accurate geometric models from rather noisy input range data and somewhat imperfect alignment.

Acknowledgments

Harry Shum is gratefully acknowledged for helpful discussions we had during the progress of this work.

References

[Curless and Levoy, 1996] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH*. ACM, 1996.

[Friedman *et al.*, 1977] Jerome H. Friedman, Jon Bentley, and Raphael Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

[Hilton *et al.*, 1996] A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt. Reliable surface reconstruction from multiple range images. In *Proceedings of the European Conference*

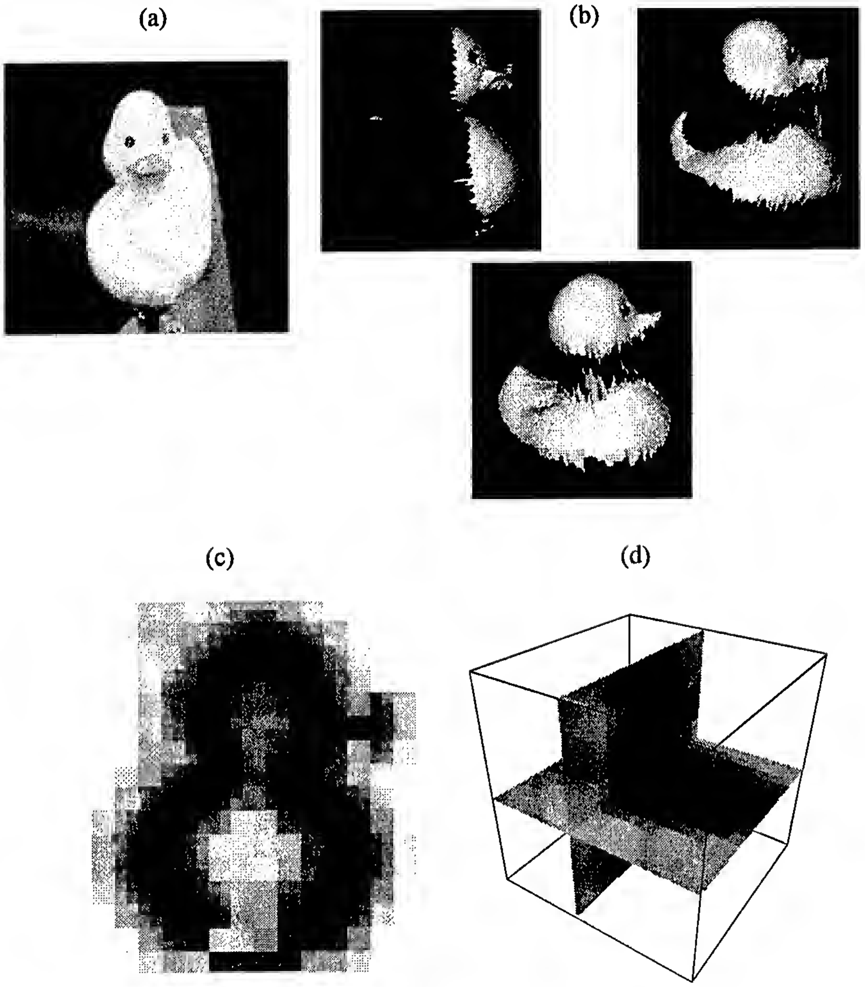


Figure 1: Results from modeling the rubber duck. (a) An intensity image of the duck, (b) a close-up of some of the triangulated range images used as input to the consensus-surface algorithm, (c) a slice of the implicit-surface octree volume where the grey-scale indicates the proximity to a surface point (black closest, white furthest), and (d) a 3D view of two cross sections of the implicit-surface octree volume.

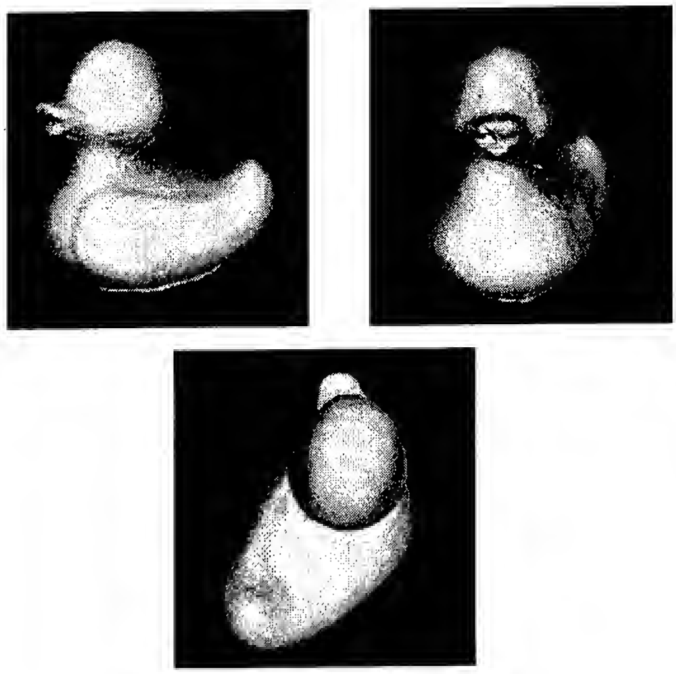


Figure 2: Three views of the resulting triangulated model of the duck.

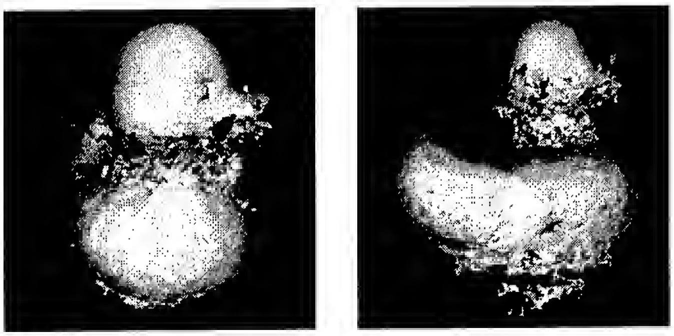


Figure 3: The result of the naive algorithm, Algorithm *ClosestSignedDistance*, on the duck image.

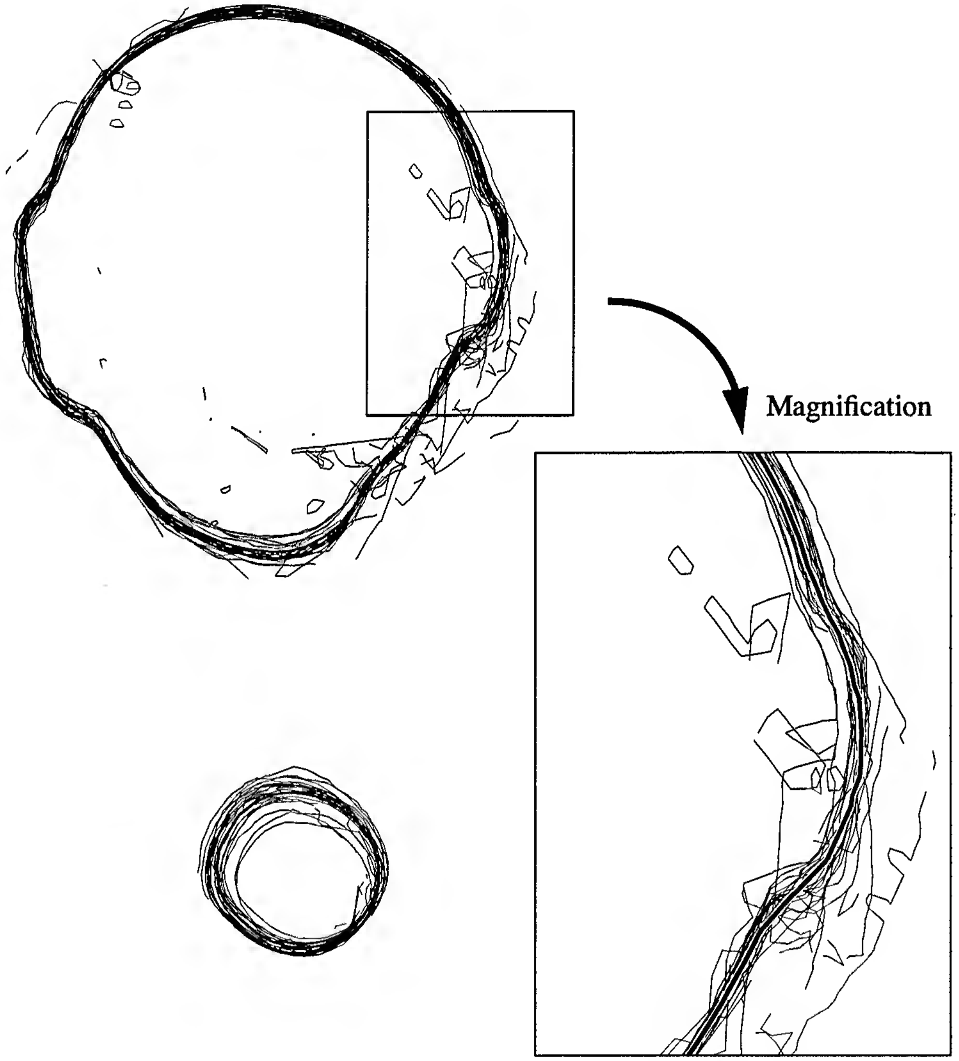


Figure 4: A cross section of the final model of the rubber duck (thick black line) and the original range-image data (thin black lines) used to construct it.

on Computer Vision, pages 117–126. Springer-Verlag, 1996.

[Hoppe *et al.*, 1992] H. Hoppe, T. DeRose, T. Duchamp, J.A. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphcics*, 26(2):71–78, 1992.

[Lorensen and Cline, 1987] W.E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH*, pages 163–169. ACM, 1987.

[Meagher, 1980] D. J. R. Meagher. *The octree encoding method for efficient solid modeling.* PhD thesis, Rensselaer Polytechnic Institute, 1980.

[Soucy and Laurendeau, 1992] Marc Soucy and Denis Laurendeau. Multi-resolution surface modeling from multiple range views. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 348–353, 1992.

[Turk and Levoy, 1994] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of SIGGRAPH*, pages 311–318. ACM, 1994.

[Wheeler, 1996] Mark D. Wheeler. *Automatic Modeling and Localization for Object Recognition.* PhD thesis, Carnegie Mellon University, 1996.

Solid Model Construction using Meshes and Volumes

Michael K. Reed and Peter K. Allen
Computer Science Department, Columbia University, New York, NY 10027

Abstract

This paper describes a method for constructing models from range data that combines both surface and volumetric modeling techniques. Each of a series of range images from different viewpoints is modeled by a mesh surface, which is then extruded to form a solid representing the volume of space occluded from the sensor from that particular viewpoint. Set intersection is used to integrate the information from different views. Results for two highly convex parts are shown. An important benefit of this method is that models can be created from a small number of views and that the modeling is done in continuous volumetric space.

1. Introduction

The use of range data to drive the modeling process has been steadily increasing over recent years. This is due not only to an increase in the availability and a decrease in price of range imaging devices, but also from an increase in the scale of use. Range data is now acquired in domains such as cartography and medicine, and is used in manufacturing processes such as inspection and in computer graphics as a method of acquiring models of real-world objects. To be useful these systems must be able to merge data acquired from multiple viewpoints to complete the modeling task. This paper describes a system that performs this task, incrementally building solid models from multiple range images. It motivates the integration of both mesh surface and volumetric models and the generation of a topologically correct 3-D model at each stage of the modeling process, allowing the use of well-defined geometric algorithms to perform the merging task. One of the key benefits of using a correct solid model at each stage in the process is that we can guarantee a water-tight model without holes, which is important for rapid prototyping and reverse engineering tasks. These solid models can also be used to automatically plan the next view during the model acquisition stage. We have been able to use our previous work in sensor planning for inspection tasks to plan the next view of an object to be modeled, thus reducing the number of scans needed to recreate a correct model [9].

2. Relation to Previous Work

Recent research on the acquisition, modeling and merging process includes the REFAB system, which allows a user to specify approximate locations of machining features on a range image of a part; the system then produces a best fit to the data using previously-identified features and domain-specific knowledge as constraints [12]. The IVIS system uses an octree to represent the seen and unseen parts of each of a set of range images and uses set-theoretic operators to merge the octrees into a final model [11]. Very recently octrees have again been used to represent range data, in combination with an isosurface extraction algorithm that produces very high resolution results, albeit from a large number of images [4]. Methods that use a mesh surface to model and integrate each of a set of range images [14][10][8] or to model a single, complete point sampling [5] have also proven useful in this task.

The popularity of mesh-based methods is due to their combination of simplicity and representational flexibility. Free-form parts may be modeled accurately without the need for multiple parametric shape representations, and they may easily incorporated into a CAD/CAM system. In addition, they lend themselves well to incremental methods that are required in situations that utilize model refinement or planning stages.

This work was supported in part by an ONR MURI Grant, DARPA AASERT awards DAAHO4-93-G-0245, DAAHO4-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

Methods that use surface models based on meshes, such as [14], however, suffer from two important problems. First, meshes by their very nature tend to introduce holes which are very difficult to remove. Creation of water-tight solids is an important end-product of these modeling systems and mesh-based methods typically require significant post-processing to remove these holes. Second, when surface meshes are joined the intersections become difficult to compute and can introduce error, particularly at places where the meshes overlap but do not actually intersect. An interesting way to alleviate these problems was developed in [4], which uses each surface mesh in a ray-casting process to weight vertices in an octree, followed by iso-surface extraction to reconstruct a closed solid. In essence, this imposes a regular volumetric structure on the irregular surface mesh. The benefit of the volumetric structure is that it allows identification of holes as well as methods to fill them. While the results of this system are quite good, there are still some issues that need to be addressed, one of which is how the system behaves with a smaller number of scans from widely varying viewpoints. Our method differs in that we represent the volume in a continuous space which precludes the need for discrete voxel structure. We construct a solid for each range image that represents the imaged surface and the space this surface occludes from the sensor. Because of this it is not necessary to explicitly represent "empty" space, and it is not necessary to consider separate operations to fill holes in the model.

The method uses a mesh to model the sensed surface of an object, and then sweeps the mesh in the imaging direction to generate a solid representation. In this regard it may be thought of as an integration of both mesh-based reconstruction techniques and work that performs edge detection and projection from intensity images [3][7]. The model created from each imaging operation is a solid topologically-correct CAD model. A benefit of this is that it allows coarse models to be created from a small number of scans that may be acceptable to some tasks, for example 3-D FAX. The method is an incremental one that allows new information to be easily integrated as it is acquired into a composite model. As more scans are merged, the model's fidelity increases. Finally, each model created by our method includes information about the volume of occlusion, which describes the space occluded from the sensor by the object, and is not present in systems that only model sensed surfaces. The occlusion volume is a key component of many sensor planning methods because it allows the system to reason about what *has not* been seen, but it has not yet been integrated sufficiently into mesh-based methods.

3. Building a Solid from A Range Image

A robotic system, comprised of a Servo-Robot laser rangefinder attached to an IBM SCARA robot, is used to acquire a range image of the object being modeled, which is placed on a motorized rotation stage (see Figure 1). The result of each scanning process is an NxM range image from a particular viewpoint, the direction of which is controlled by turntable rotation. The 3D point data from the range image are then used as vertices in a mesh surface model.

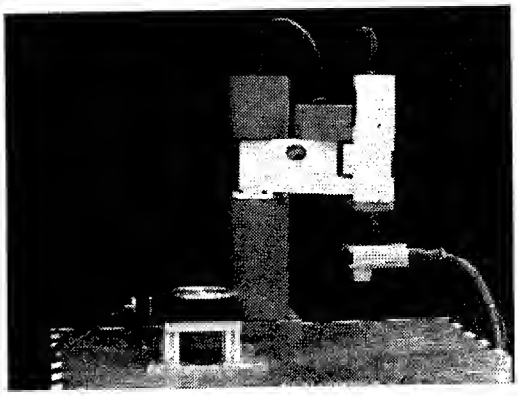


Figure 1. Experimental setup showing robot with attached laser rangefinder (to right) and turntable (to left).

3.1 Sweeping a Mesh to Extrude a Solid Model

Incremental modeling techniques that use a mesh surface model can not form a closed solid until a set of images that completely cover the surface of the object has been acquired. The primary disadvantage of this is that it prevents the use of a planning method or any other procedure that requires a solid model. The second disadvantage is that topological information present in each image is not retained since all intermediate modeling steps are represented by non-manifold surfaces. A solution to this problem is to "sweep" the mesh surface to extrude a solid model of both the imaged object surfaces and the occluded volume. This method may be stated concisely as:

$$S = \bigcup_{\forall i} extrude(M_i)$$

Each triangular mesh element M_i is swept orthographically along the vector of the rangefinder's sensing axis until it comes in contact with a far bounding plane, resulting in the 5-sided solid of a triangular prism (Figure 2). A union operation is then applied to the entire set of prisms, which produces a polyhedral solid consisting of three sets of surfaces: a mesh-like surface from the acquired range data, a number of lateral faces equal to the number of vertices on the boundary of the mesh derived from the sweeping operation, and a planar bounding surface that caps one end.

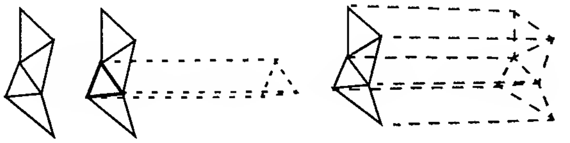


Figure 2. Example of a mesh sweep operation. (left to right) Mesh surface, mesh surface with one element swept, and mesh surface with all elements swept and unioned. The sensing direction is from the left.

This method is particularly simple to implement because it uses two operations that are available in virtually every 3D CAD package: linear sweep of a triangle with no draft angle, and regularized intersection of axis-aligned and adjacent prismatic solids.

4. Merging Single-View Models

Each successive sensing operation will result in new information that must be merged with the current model being built. Merging of mesh-based surface models has been done using clipping and re-triangulation methods. These methods are necessary because these meshes are not closed, and because of this specialized techniques to operate on non-manifold surfaces of approximately continuous vertex density were needed. In our method we generate a solid from each viewpoint which allows us to use a merging method based on set intersection. Many CAD systems include highly robust algorithms for set operations on solids, and our algorithm takes advantage of this. This is of critical importance in this application for the following reasons: the high density of the range images (and therefore the small size of many of the mesh elements), the many long and thin lateral surfaces, and most importantly the fact that many of these models will have overlapping surfaces that are extremely close to each other.

The merging process itself starts by initializing the sensed "composite" model to be the entire bounded space of our modeling system. The information determined by a newly acquired model from a single viewpoint is incorporated into the composite model by performing a regularized set intersection operation between the two.

5. Experimental Results

To show the behavior of this system, the reconstruction of two parts is shown. For both parts, a small number of range images were taken, using turntable rotation to change the sensor viewpoint. The solids constructed from these images are shown, along with the final model constructed from merging of those solids. The final models would then be prime candidates for a decimation algortihm, such as the one presented in [2].

The first part is the plastic hand controller (shown in Figure 3) for a home video game machine: it consists of polygonal and curved surfaces at varying levels of detail, including various buttons on its front surface. Three range images were take of this part, and swept into the solids shown in Figure 4.. The result of the

Figure 3. Photograph of the video game controller.

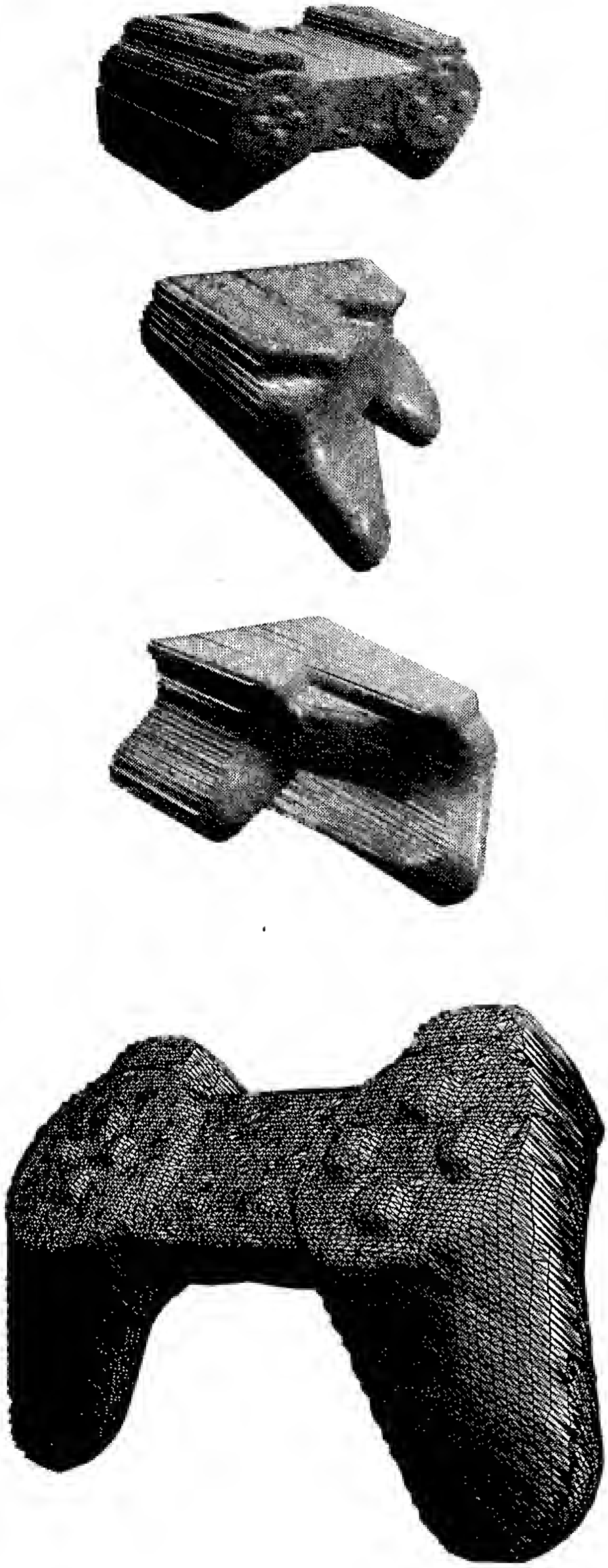


Figure 4. Top: solid models constructed from each of three range images of the game controller, each 120 degrees apart. Bottom: the final model.

intersection is shown at the bottom of Figure 4. This model has been used as input to a rapid prototyping machine and the model has been physically built.

The second model is that of a strut-like part, which consists of curved and polygonal surfaces, large self-occlusions, and two through-hole features, as shown in Figure 5. This model for this part was constructed with the assistance of our sensor planning system to plan views for hand-designated targets [9]. In this case, four images were taken at irregular angular separations, the models of which are seen in Figure 6. The final model is shown in Figure 7, and captures the outer surface and one of the two through-holes well.

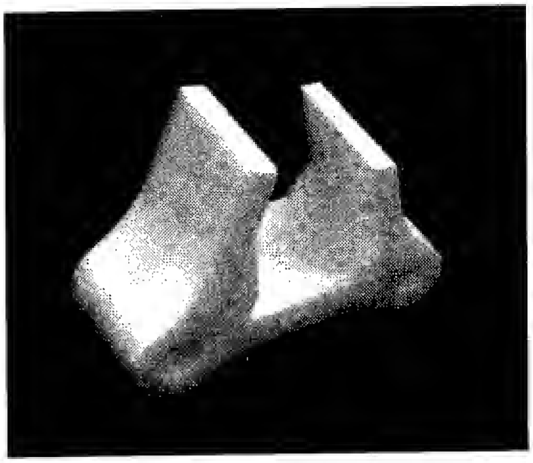


Figure 5. Photograph of the strut part.

6. Discussion

Although this method is simple and effectively builds many parts using only a few sensing operations, there are two issues that make the reconstruction process difficult. The first problem is that of determining effective next viewpoints, which is by no means a problem specific to our modeling method. The second problem is due to the behavior of set intersection on models built from sampled data, and here it is necessary to pay more attention to this detail than is required in other methods.

To effectively determine the next viewpoint for a sensing operation, it is necessary to combine a planning component with the modeling system. The planning component we are integrating into our modeling system [9] is based on previous work on the sensor

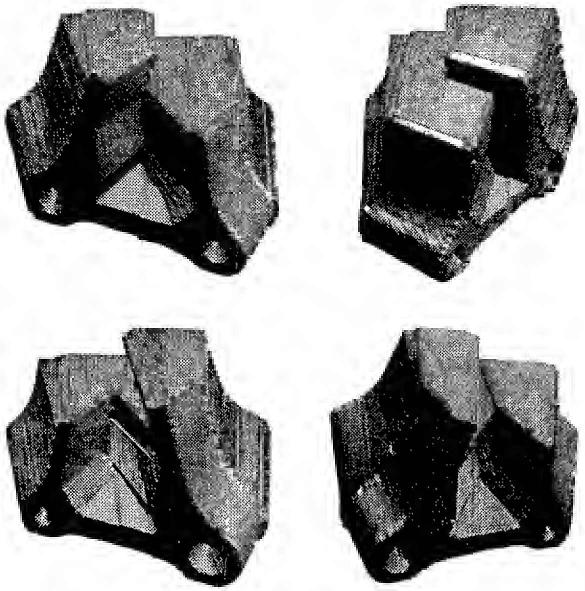


Figure 6. Four models of the strut part, with uneven rotations between each view.

planning problem in our laboratory [1][13]. The sensor planning problem is that of computing a set of sensor locations for viewing a target given a model of an object or scene, a sensor model, and a set of sensing constraints. The planner is used to reason about occlusion to compute valid, occlusion-free viewpoints given a specific surface on the model. Our system labels each surface as a sensed model surface or one caused by the boundary with the occluded volume.Using this information, we can plan sensor positions for the next view that reduce the occluded volume but do not suffer from model self-occlusion. Once an unoccluded sensor position for the specified surface has been determined, it may then be sensed, modeled, and integrated with the composite model. Thus, the method is target-driven and performed in continuous space. As the incremental modeling process proceeds, regions that require additional sensing can be guaranteed of having an occlusion free view from the sensor if one exists. Other viewing constraints may also be included in the sensor planning such as sensor field of view, resolution, and standoff distance.

The second problem described above is more specific to the use of set intersection as a tool for volumetric integration. In all systems it is assumed that the object's surface is "well behaved" with respect to the distance between sample points. What this means is that for the object's surfaces that are visible to the sensor and that are not highly inclined, the deviation from the mesh element between adjacent samples is small. Although the sampling interval depends on the distance to the target, these values are typically ~2-3mm for a perpendicular surface. However, there may be surfaces that are highly inclined to the sensor, which results in large distances between samples, and therefore the possibility of a large deviation of the object surface from the mesh element between samples. Mesh-based methods that use only a surface model handle this problem by disregarding these data, with the assumption that the surface there will be acquired in a later imaging operation from another viewpoint. In our method, however, this can have a very detrimental effect on the final model.

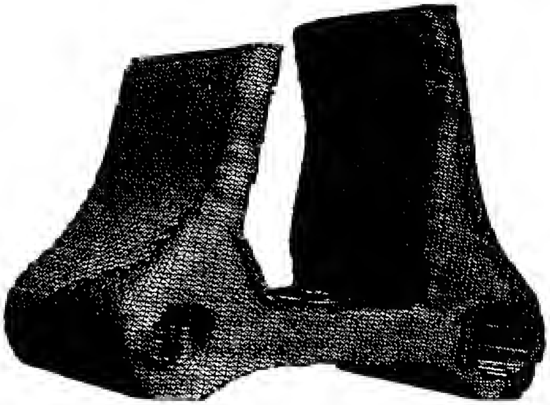


Figure 7. The final model of the strut.

To see why this is so, consider the 2-D example in Figure 8., which represents one scan line from the rangefinder. Due to the subsampling of the scene, an inappropriate mesh surface (in 2-D, this is a line segment) will be created that passes through the interior of the object's true geometry. In modeling systems such as ours that rely on set intersection as a means of integration, this is unacceptable because once a set of points has been classified as "outside" the model, there is no way to recapture that information no matter how many subsequent images correctly include those points.

One possible solution to this problem would be dilate the model at those surfaces where the problem occurs. It is clear from Figure 8 that the inappropriate

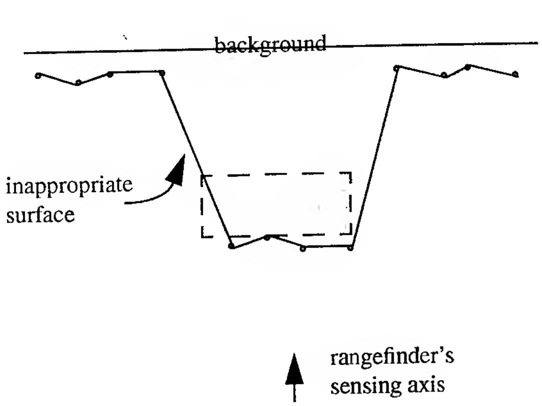


Figure 8. 2-D example of typical surface error caused by constructing mesh from sampled data points.

surfaces occur at extremal boundaries in the scene, which correspond to mesh elements for which the surface normal and sensing direction vectors are nearly perpendicular. It should be possible then to dilate the mesh at these surfaces to counteract the effects of subsampling. Because these surfaces will always require more sensing in any case, this dilation does not affect any sensing strategy. We are currently exploring this and volume tolerancing as solutions to this problem.

7. Conclusions

We have described a system that builds a 3-D CAD model of an object incrementally from multiple range images. It motivates the generation of a solid model at each stage of the modeling process, allowing the use of well-defined geometric algorithms to perform the merging and integration task. We have been able to create models of a number of different objects using just three or four scans. As we increase the number of scans, the model fidelity will also increase. A benefit of our method is that the volumes created can be used in conjunction with our previous work in sensor planning to plan the next view. Finally, we are refining this system to improve the fidelity of the models to alleviate problems caused by using set intersection as a tool for volumetric integration.

8. References

[1] Steven Abrams, Peter K. Allen, and Konstantinos A. Tarabanis. Dynamic Sensor Planning. In *Int. Conf. on Intelligent Autonomous Systems*, pages 206–215, Pittsburgh,PA,1993

[2] J Cohen at al., Simplification Envelopes. In *Proceedings of SIGGRAPH*, pp.119-128, 1996.

[3] C. Connolly and J. Stenstrom. 3d Scene Reconstruction from Multiple Intensity Images, in *Proceedings of the IEEE International Conference on Robotics and Automation*, p. 124-130, May, 1989.

[4] B. Curless and M. Levoy. *A Volumetric Method for Building Complex Models from Range Images.* In *Proceedings of SIGGRAPH*, p. 303-312, 1996.

[5] H.Hoppe. *Surface Reconstruction from Unorganized Points*. Ph.D. thesis, Dept. of Com. Science and Engineering, U. of Washington, 1994.

[6] K. Ikeuchi and P.J. Flynn, Editorial: Recent Progress in CAD-based Vision, Computer Vision and Image Understanding, 61(3), May 1995.

[7] W. M. Martin and J. K. Aggarwal. Volumetric Descriptions of Objects from Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[8] R. Pito, Mesh Integration Based on Co-Measurements, In *Proceedings of 1996 IEEE International Conference on Image Processing*, Lausanne, Switzerland.

[9] M. Reed, P. Allen, and I. Stamos, 3-D modeling from range imagery: An Incremental Method with a Planning Component, To appear in *Proceedings of the International Conference on Advances in 3-D Digital Imaging and Modeling*, Ontario, Canada, 1997.

[10] M. Rutishauser, M. Stricker, and M. Trobina. Merging Range Images of Arbitrarily Shaped Objects. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 573–580, 1994.

[11] G.H. Tarbox and S.N. Gottshlich. Ivis: An Integrated Volumetric Inspection System. *Computer Vision and Image Understanding*, 61(3):430–444, may 1995.

[12] W.B. Thompson, H.J. de St. Germain, T.C. Henderson, and J.C. Owen. Constructing High-Precision Geometric Models from Sensed Position Data. In *Proceedings 1996 ARPA Image Understanding Workshop*, pages 987–994, 1996.

[13] K. Tarabanis, R. Tsai, P.K. Allen. The MVP Sensor Planning System for Robotic Tasks. In *IEEE Trans. on Robotics and Automation*, 11(1), p. 72-85, February 1995.

[14] G. Turk and M. Levoy. Zippered Polygon Meshes from Range Images. In *Proceedings of SIGGRAPH*, pp. 311–318, 1994.

Constraint Optimization and Feature-Based Model Construction for Reverse Engineering

H. James de St. Germain, Stevan R. Stark,
William B. Thompson, and Thomas C. Henderson
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
http://www.cs.utah.edu/projects/robot/sam.html

Abstract

Reverse engineering techniques can be used to aid in the manufacture of spare parts when original parts inventories are exhausted. For mechanical parts, this process involves sensing the geometry of an existing part, creating a geometric model of the part from the sensed data, and then passing this model to an appropriate CAD/CAM system for manufacturing. Sensing errors and the modeling demands of modern CAD systems present serious challenges to the reverse engineering process. We describe constraints that can be used to simplify the process and aid in the construction of higher quality models than would otherwise be possible.

1 Introduction

The DOD must maintain a large quantity of legacy hardware, much of it decades old. Part inventories are frequently exhausted well before decommissioning of the relevant pieces of equipment and additional spares are often difficult or impossible to obtain from the original suppliers of the equipment. *Reverse engineering* techniques can be used to create CAD models of a part based on sensed data acquired using three-dimensional position digitization techniques, allowing the manufacture of new spare parts based on an analysis of existing parts, without the need for original CAD models or other documentation [Traband *et al.*, 1996]. This process involves organizing and editing sensed 3–D positions and then fitting surfaces to these points in a manner such that the resulting geometric models can be imported into a CAD system. High precision in the modeling process is often required if the final re-engineered parts are to be usable. The sensors used to gather data are subject to a wide variety of random and systematic errors, adding to the difficulty of constructing precise models.

Modeling accuracy depends on effective use of properties that distinguish the geometry of interest from effects due to sensing errors. In the case of geometric modeling of man-made objects, it is often critical to recognize that certain shapes are more likely than others and to use this information to reduce the inherent degrees of freedom of the modeling primitives used to fit the data (Figure 1).

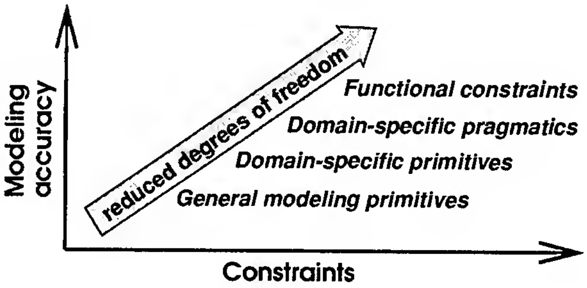

Modeling accuracy
reduced degrees of freedom
Functional constraints
Domain-specific pragmatics
Domain-specific primitives
General modeling primitives
Constraints

Figure 1: Constraints reduce degrees-of-freedom and increase modeling accuracy.

Three important class of constraints aid the reverse engineering of mechanical parts [Thompson and Henderson, 1996]. *Domain-specific primitives* reconstruct part geometry using the same modeling primitives available to the original part designer [Owen *et al.*, 1994, Thompson *et al.*, 1995, Thompson *et al.*, 1996]. *Domain-specific pragmatics* capture common design practice. *Functional*

This work was supported by the Advanced Research Projects agency under Army Research Office grant number DAAH04-93-G-0420.

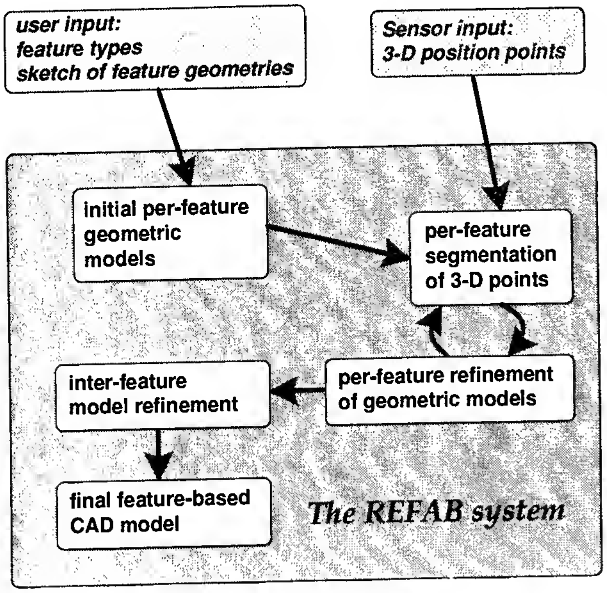

user input:
feature types
sketch of feature geometries
Sensor input:
3-D position points
initial per-feature geometric models
per-feature segmentation of 3-D points
inter-feature model refinement
per-feature refinement of geometric models
final feature-based CAD model
The REFAB system

Figure 2: Steps in the model construction process.

constraints capture aspects of the likely intent of the design. Operationally, we exploit these constraints by utilizing three types of information in our model creation process: geometric primitives that are capable of describing object geometry with minimal parameters, local constraints which when possible use a restricted set of sub-primitives to represent geometry, and global constraints based on how different primitives interact in a properly designed object.

Figure 2 shows the various steps involved in the reverse engineering process that we have developed. A user provides a rough description of the manufacturing features making up the part, which is used to create an initial geometric model. This model is used to segment sensor data into position points corresponding to each features. Modeling primitives are fit to the segmented data. The fitting and segmenting process can be iterated to refine the models. Finally inter-feature constraints are used to adjust the individual feature descriptions to better approximate a globally optimal description of the part.

2 Fitting Per-Feature Models.

To demonstrate the effectiveness of feature-based reverse engineering, we have created a prototype system called REFAB (**R**everse **E**ngineering – **F**e**A**ture-**B**ased). In this section, we outline the processes involved in fitting profile pockets, the most complex feature currently handled by the system, to sensed data. A profile pocket is a $2\frac{1}{2}$–D feature consisting of a subtractive volume defined by an arbitrary closed planar contour which is swept through a specified distance along a perpendicular axis.

2.1 Data Segmentation

Most segmentation methods applicable to 3–D point data involve bottom-up processing, where data is grouped based on planar surfaces for polyhedra, or on orientation discontinuities for curved surfaces [Besl and Jain, 1988, Suk and Bhandarkar, 1992]. Since few machined parts are naturally represented as polyhedral models and methods involving orientation discontinuities are problematic with noisy data, we utilize a more robust top-down segmentation approach.

The segmentation of the 3–D points corresponding to a particular profile pocket starts with the user sketching the profile contour and indicating points on the object at the bottom of the pocket and on the surface into which the pocket is cut. Planes are fit to the selected bottom and top points using the least median squares (LMedS) technique [Rousseeuw and Leroy, 1987]. Next, planes are fit to *all* data points consistently close to these initial estimates. The estimates allow the use of an efficient trimmed distribution least-squares method without a significant increase in the sensitivity to outliers [Thompson *et al.*, 1995]. The resulting planes serve two purposes: they define the sweep extent of the pocket and they provide an estimate for the axis along which the profile contour is swept.

Once the orientation of the pocket is known, the user sketch of the pocket profile can be projected onto a 2–D plane. User specified points are used to construct an initial Bezier curve approximation to the profile side [Schneider, 1990]. This in turn forms the starting point for the segmentation of 3–D position points corresponding to the sides of the profile. The full point cloud is examined for points satisfying three criteria:

- **Distance check.** Selected points must be close to the profile curve along a direction perpendicular to the sweep axis of the pocket.
- **Bounding plane check.** Selected points must be within the bounding planes specifying the sweep extent of the pocket.
- **Normal direction check.** Estimated surface

normals for selected points must be approximately perpendicular to the sweep axis of the pocket and close to the orientation of the nearest point on the Bezier curve.

Selected points likely to correspond to the profile side are used to fit the profile. If necessary, the new profile can be used to produce a better segmentation, iterating the process.

2.2 Feature Parameterization and Fitting

The previous steps result in a geometric model specified in terms of splines and bounding planes. This is as far as most reverse engineering systems proceed. Further simplifications are possible, however, by recognizing that it is common design practice to specify profiles in terms of lines and arcs rather than totally free form curves. We exploit this pragmatic by converting the Bezier curve representation of a profile to a line/arc formalism whenever this adequately approximates the data. The result is a model that is more likely to fit the original design, though the data fitting errors will be larger.

Our current implementation allows for line segments interspersed with one, two, or three consecutive constant radius arcs. The entire profile is presumed to be G1 (tangent) continuous. Triple arcs are limited to a special reinforcing feature (boss) which is symmetric. Because we have a good estimate of the sweep direction of the feature, all analysis can be done in 2–D by projecting segmented points corresponding to the side of the pocket into a plane perpendicular to the sweep direction.

Figure 3 outlines the process. First, the data points are ordered by projecting them onto the spline approximation of the profile. Then, standard polyline approximation algorithms can be used to represent the data as a sequence of line segments [Jain *et al.*, 1995]. Long segments in the polyline are presumed to correspond to line segments in the original design. To improve the accuracy with which these line segments are described, they are re-fit using a trimmed distribution least-squares method similar to that used for plane fitting.

1. Construct polyline approximation to profile.
2. Refit long line segments to data
3. Identify corners
4. Identify single, double, and triple arcs
5. Fit remaining data with splines.

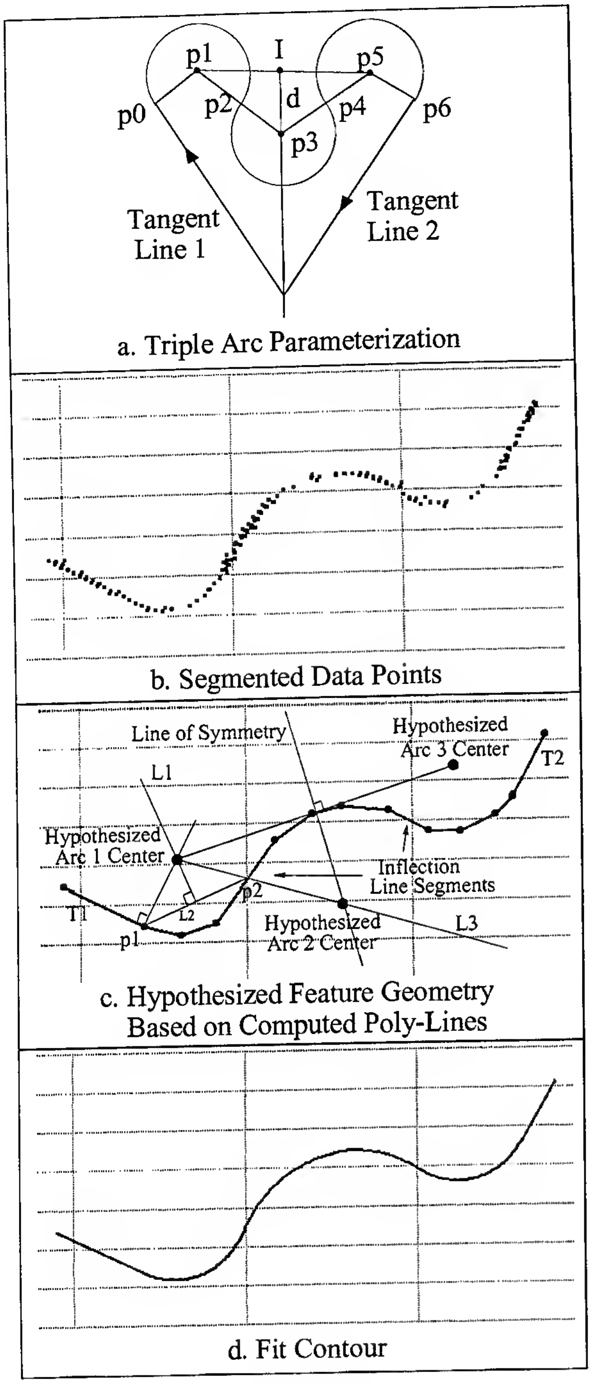
Figure 3: Contour Fitting

Non-contact position digitizers smooth data near sharp corners. As a result, the existence of corners must be inferred when the polyline has two nearly adjacent long line segments with distinctly different orientations. In such cases, the line segments are extended to form a true corner. The resulting polyline consists of long segments corresponding to true straight lines in the original profile and sequences of shorter line segments corresponding to curves.

An attempt is made to fit each of these curve sections with sub-features corresponding to a single, double, or triple arc. The dimensionality of the curve fitting can be substantially reduced by taking advantage of tangent continuity. A generic single arc has five degrees of freedom: the center point, two angles, and a radius. Tangent continuity reduces this to one, a radius. A double arc can be modeled using only four parameters. The symmetric triple arc has only three degrees of freedom. If the data cannot be adequately approximated with any of these sub-features, it is assumed to be a free form curve and is approximated with a spline.

2.3 Example: Fitting a Triple Arc Sub-Feature

In the case of the symmetric triple arc, we use a parameterization specifying the center of the first arc and the offset of the center of the middle arc from a line connecting the centers of the two outside arcs (Figure 4.a). Together with the tangent continuity constraints deriving from the adjacent line segments, this fully specifies the configuration. Fitting arc sub-features requires non-linear optimization. Both performance and the quality of the final result depend on the availability of a good initial guess. For triple arcs, the line segments which form inflections in the polyline are found first (Figure 4.c.) L2 is produced by building a line segment between $p1$ (the end of tangent line T1) and $p2$ (the mid point of the first inflection segment.) The center of arc one is hypothesized by intersecting the perpendicular bisector of L2 and the line perpendicular to T1 at $p1$. The center of arc two is generated by



a. Triple Arc Parameterization

b. Segmented Data Points

c. Hypothesized Feature Geometry Based on Computed Poly-Lines

d. Fit Contour

Figure 4: Triple arc sub-feature parameterization, initialization, and fitting.

intersecting the line L3 with the line of symmetry. The final arc is symmetric to the first.

These guesses are used to instantiate the initial parameterization for the triple arc fitting process. This fitting process is a non-linear optimizer which perturbs the parameters until no further perturbations reduce the total error from the data to the feature. Figure 4.d shows the result of the final fit. Similar techniques are used for the other arc features.

3 Optimizing Constrained Feature-Based Models

Reconstructing models of free form geometrical surfaces is a difficult process, requiring the instantiation of large numbers of parameters. Fitting real world data complicates this process by introducing problems of occlusion and noise. By asserting functional constraints over the geometry, the number of parameters can be greatly reduced and the susceptibility to noise lowered. In addition, the resulting model can be made to conform to preconceived ideas about standard $2\frac{1}{2}$–D geometry used in manufacturing.

In much the same way as an understanding of the manufacturing process suggests a feature-based approach to reverse engineering, additional understanding of the manufacturing and design processes suggests that features are not encountered in isolation, and thus certain geometric properties should hold over features. We believe that an understanding of the design process can yield large benefits in the ability to create accurate models. Feature-based models provide modeling primitives and can enforce low-level constraints over these primitives, reducing the number of variables necessary to represent an object. Constraint based techniques apply high-level constraints over these features, further reducing the DOFs as well as enforcing hypothesized design intent.

The geometry found in most manufactured parts is based on simple and consistent rules [Hsu, 1996]. Such parts are usually designed in the most straightforward way, using common *design practices*. This motivates the idea that apparent relations between features are likely to be by design and not by coincidence. Some modern design systems are starting to enforce this idea by restricting the designer to a set of design features (hierarchical structures related to design intent.) We utilize this information by enforcing constraints during the reverse engineering process. For example, if two planar surfaces on a part *appear* parallel, then logic dictates that the designer intended them to be parallel and they are fit as such.

Consider the feature-based model shown in Figure 5. An engineer, given a part milled from this

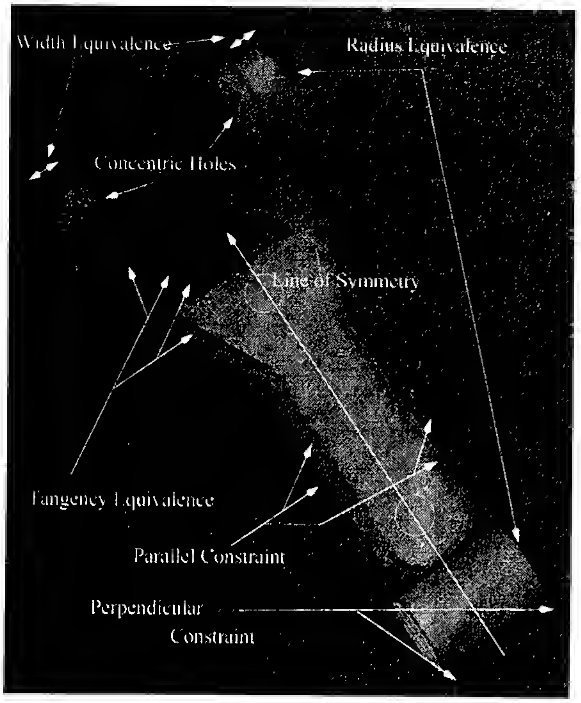


Figure 5: Constraints on Lower Link Model

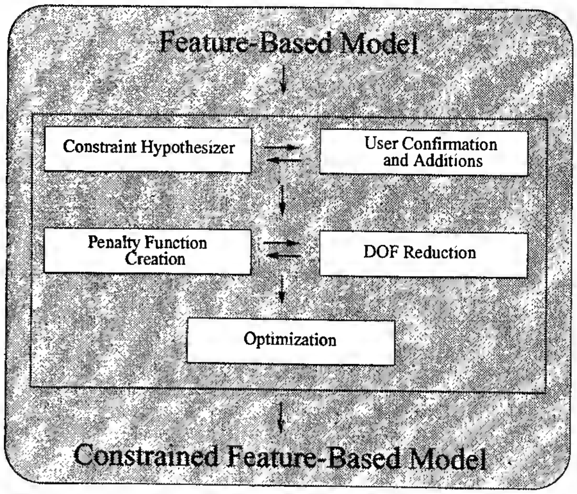


Figure 6: Constrained Optimization Process

High Level	**Complex**
Parallelism	*Symmetry*
Perpendicularity	*Tangent Equivalence*
Concentricity	*Identical but Offset*
Radius Equivalence	
Width Equivalence	

Figure 7: Supported Constraints

model, would no doubt recognize a simple 2–D profile, two weight reduction pockets, and several through holes. Further inspection might lead the engineer to hypothesize certain geometric constraints that are likely to hold for this type of part: the part seems symmetric, the larger holes seem to be the same size (two seem to be concentric), the inner pocket seems to be aligned with the outer profile, etc. Figure 5 displays these and other constraints. By enforcing these properties during model creation, the final result is more likely to accurately represent the intended geometry of the part.

The advantages of constraint based model fitting are threefold. First, the resultant model is more likely to capture the design intent of the part. Secondly, the number of degrees of freedom necessary to fit the part can be significantly reduced, increasing the speed and accuracy of the fitting process and reducing the sensitivity to noise. Finally, features on the part which are hardest to sense or perhaps contain the greatest degradation can often be constrained to other features which correspond to more valid data. We call this process *accuracy transference*.

3.1 The Constrained Optimization Process

Modeling accuracy can be increased by recognizing constraints between feature primitives. The constrained optimization process is initialized with a feature-based model (Figure 6). The following three steps are then applied. First, a hypothesis step attempts to semi-automatically identify possible constraints on the model. Second, a new model is formed which reduces the number of parameters necessary to represent the object. During this stage, all constraints which are not geometrically represented are expressed by penalty functions. Finally, an optimization process, utilizing a weak search algorithm, is used to adjust the parameters in order to produce the best model which fits the data while obeying the constraints.

3.1.1 Constraint Hypothesis

An understanding of the design process suggests certain practices which can be directly described by geometric constraints. We list the $2\frac{1}{2}$–D constraints we have identified and implemented in Figure 7. Radius and width equivalence apply to distance measures. Tangent equivalence dictates that intersecting sub-features, such as lines and arcs, are

G1 continuous. Identical but offset features represent geometrically identical features which are transformed into separate frames. Parallelism and perpendicularity are applied both to 2–D line segments and to 3–D planar surfaces. The other constraints have standard geometric interpretations.

Currently a semi-automatic constraint finding algorithm is used. The algorithm hypothesizes possible constraints based on parametric equivalence. For instance, consider the 2–D projection of two holes onto the same anchor plane. If the centers of these two holes are located within some small distance, ε_1, from each other, then there is high likelihood that the two holes are concentric and the fit should constrain them as such. Likewise, if two profile lines intersect with an angle of 90 degrees +/- ε_2, then a perpendicular constraint is likely, and should be asserted. The algorithm cross checks all features and sub-features versus all other features and sub-features looking at all appropriate constraints. Tangency constraints are initially asserted for all profile contours.

Once the system has asserted all likely constraints, the user is requested to verify them and is further allowed to add new constraints. This is necessary because the hypothesis process can fail in two instances. First, if the original fit is off, then parametric equivalence is unlikely to be satisfied. This can be the case when the part was poorly manufactured or fixtured, or when the sensing process cannot acquire valid data. Second, complex constraints such as symmetry are not directly inferred from the model and require manual intervention. In our test cases, the hypothesis process has been able to identify most valid constraints, with less than a 5% false positive ratio.

3.1.2 DOF Reduction and Penalty Functions

Once the constraints on the part have been specified, the process generates a new model for the part based on these constraints and the initial feature-based model. First, the parameter set is reduced using degree of freedom reduction. Second, any constraints which cannot be represented by a simple parameter reduction are implemented by building penalty functions to represent them. The resulting model includes a reduced parameterization and a set of penalty functions associated with the constraints. This new model is used to drive the optimization process.

DOF reduction is the symbolic substitution of parametric variables representing the same values. For example, a hole is represented by its center and radius. If the hole is concentric to another hole, then a symbolic substitution is made so that any reference to the second hole's center is replaced by the first hole's center. Likewise, two parallel lines, using polar coordinates, can use the same symbolic variable for their theta parameter. Unfortunately, symbolic substitution does not work when the parameterizations between two features are not compatible or when this would over constrain the model. For example, consider a single arc concentric with a triple arc. The center of a single arc sub-feature is computed using its tangent lines and the radius of the arc. The center of a triple arc feature is computed based on its tangent lines and three non-intuitive parameters. Both parameterizations imply a center but do not explicitly contain that information. Thus to symbolically try to equate the two features' centers would be meaningless. Therefore, another way must be used to represent these constraints.

Such constraints can be expressed using penalty functions. Penalty functions are a class of functions which equate to zero when a condition is met, but rapidly increase in value as the condition decreases in validity. Thus, the penalty function, ϕ, associated with two concentric holes might be:

$$\phi(p_1, p_2) = \mathcal{K} * \delta(p_1, p_2)$$

where p_1 and p_2 are the centers of hole one and hole two, respectively, δ computes the Euclidean distance between two points, and $\mathcal{K}$ is a non-negative constant. By setting $\mathcal{K}$ to zero, the constraint is ignored; by setting $\mathcal{K}$ to an arbitrarily large value, the optimization process strictly enforces the constraint. Optimization theory details the use of penalty functions and the use of multiple $\mathcal{K}$ values [Gottfried and Wiesman, 1973]. By using an exterior-point algorithm we are able to start our search outside the feasible area and then move into it, guided by the data fit and constraint penalties.

3.1.3 Optimized Fitting

A generic optimization problem involves following a multi-dimensional function down hill until a global (or local) minimum is found. This is accomplished by iterating over a list of parametric values, perturbing them until no new change leads to a lower value. Once a minimum is found, the search routine is generally reinitialized with the computed values and restarted in an attempt to escape local minima. Refer to [Press *et al.*, 1991] for a discussion on the Amoeba and Powell methods, which we currently use.

The error function to be optimized is based on the distance from the sensed data to the generated model plus the penalties associated with each constraint penalty function.

$$\textit{model error} = \sum_{i=0}^{f} \mathcal{S}_i * \sum_{j=0}^{p} \delta(pt_j, F_i)$$

$$\textit{penalty error} = \mathcal{K} * \sum_{i=0}^{P} k_i * \mathcal{P}_i$$

Where f is the number of features, $\mathcal{S}$ is the precedence associated with the current feature, j is the number of points for feature i, δ is the distance from the data point to the feature, and F is the set of features. $\mathcal{K}$ is the penalty function multiplier, P is the number of penalty functions, k is a scaling constant, and $\mathcal{P}$ is the set of penalty functions. The sum of the model error plus the penalty error is used to drive the optimization process.

During each iteration of the search, the value $\mathcal{K}$ is gradually increased in order to enforce the constraints without adversely restricting the search. The scaling factors k_i transform the various penalty measures into one meaningful level of magnitude. The precedence value $\mathcal{S}$ gives the engineer a way to weight the data pertaining to each feature, and thus *transfer* accuracy from features fit with presumed accurate data to features where data is unreliable.

The penalty function multipliers must be chosen so as to enforce the constraints without removing the impact of the data on the fit. Thus, as the constraints are more rigidly enforced, the associated error to the data will increase while the error to the intended model will decrease.

4 Results

We have applied our reverse engineering techniques to several $2\frac{1}{2}$–D parts from the Utah benchmark suite [Thompson and Owen, 1994]. The benchmark suite provides high quality milled parts and their CAD models, allowing us to produce quantitative analysis of the reverse engineering process. Each part was painted in order to avoid specularity problems, and then scanned from multiple views using a non-contact laser digitizer. Refer to [Thompson *et al.*, 1995] for more detail on the sensing process. The resultant unordered 3–D point clouds were registered into a single coordinate system based on common planar surfaces [Shum *et al.*, 1994].

Results wrt. Models	Constraints Used				
	None	Local		Global	
	DOF	DOF	RMS	DOF	RMS
Lower Link					
outer profile	54	22	37	22	32
pocket one	42	16	148	5	117
pocket two	42	16	263	5	117
planar surfaces	12	12	30	6	6
overall	150	66	**95**	38	**48**
Shock Plate					
outer profile	21	9	36	9	40
pocket one	51	15	179	12	146
pocket two	51	15	176	0*	146
hole one	3	3	123	1	124
hole two	3	3	93	1	63
hole three	3	3	79	1	71
planar surfaces	12	12	39	6	21
overall	144	60	**86**	30	**72**
Steering Arm					
outer profile	25	13	51	11	38
hole	3	3	55	1	44
planar surfaces	6	6	32	4	32
overall	34	22	**47**	16	**36**

* Identical but offset to the top pocket feature.

Table 1: Difference between recovered model and original model (in microns).

The combined data cloud for each part, was reverse engineered using the REFAB system. This output was utilized to instantiate the constraint process. Both methods produce CAD models compatible with the University of Utah's Alpha_1 CAD/CAM system [Drake and Sela, 1989]. The results shown in Table 1 represent the error associated with each part and the degree-of-freedom reduction accomplished. Precision was seen to improve by a factor of 1.2 to 2.0 based on the number of constraints asserted on each part. The error measure represents the RMS value of the distance from a uniformly sampled distribution of points on the reverse engineered model to the original Alpha_1 model. This value was calculated using standard CAGD methods.

5 Conclusion

Most model reconstruction techniques apply generic primitives to a wide range of problems. We use domain-specific primitives and information on a narrow set of objects, in order to reconstruct high precision models. Three levels of abstraction are used. First, we use features to describe the geometry of the object with minimal degrees-of-freedom. Second, we apply local constraints to these features, further reducing the DOFs. Finally, we apply global constraints to features and sub-features, once again reducing the DOFs associated with the object while also attempting to capture the original design intent.

Domain-specific reconstruction techniques require detailed information of the area of interest, but provide substantial benefits in the accuracy and usefulness of the recovered model. We have quantitatively shown their utility in the realm of $2\frac{1}{2}$–D milled parts and have described in detail, the underlying techniques used by our system to segment data, fit feature-based models, and to apply constraints to these models.

References

[Besl and Jain, 1988] P. Besl and R. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 167–192, March 1988.

[Drake and Sela, 1989] S. Drake and S. Sela. A foundation for features. *Mechanical Engineering*, 111(1), January 1989.

[Gottfried and Wiesman; 1973] B.S. Gottfried and J. Wiesman. *Introduction to Optimization Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[Hsu, 1996] C-Y. Hsu. *Graph-Based Approach for Solving Geometric Constraint Problems*. Ph.D. dissertation, University of Utah, 1996.

[Jain *et al.*, 1995] Ramesh Jain, Rangachar Kasturi, and Brian G. Shunk. *Machine Vision*. McGraw-Hill, 1995.

[Owen *et al.*, 1994] J. C. Owen, P. P. J. Sloan, and W. B. Thompson. Interactive feature-based reverse engineering of mechanical parts. In *Proc. ARPA Image Understanding Workshop*, pages 1115–1124, November 1994.

[Press *et al.*, 1991] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge, Cambridge, MA, 1991.

[Rousseeuw and Leroy, 1987] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[Schneider, 1990] P. J. Schneider. An algorithm for automatically fitting digitized curves. In A. S. Glassner, editor, *Graphics Gems*. Academic Press, 1990.

[Shum *et al.*, 1994] H. Y. Shum, K. Ikeuchi, and R. Reddy. Virtual reality modeling from a sequence of range images. In *Proc. ARPA Image Understanding Workshop*, pages 1189–1198, November 1994.

[Suk and Bhandarkar, 1992] M. Suk and S. Bhandarkar. *Three Dimensional Object Recognition from Range Images*. Springer-Verlag, New York, 1992.

[Thompson and Henderson, 1996] W. B. Thompson and T. C. Henderson. IU at the University of Utah: Building 3-D models from sensed data. In *Proc. ARPA Image Understanding Workshop*, pages 205–210, February 1996.

[Thompson and Owen, 1994] W. B. Thompson and J. C. Owen. "Hard-copy" benchmark suite for image understanding in manufacturing. In *Proc. ARPA Image Understanding Workshop*, pages 221–227, November 1994.

[Thompson *et al.*, 1995] W. B. Thompson, J. C. Owen, and H. J. de St. Germain. Feature-based reverse engineering of mechanical parts. Technical Report UUCS-95-010, University of Utah, July 1995.

[Thompson *et al.*, 1996] W. B. Thompson, H. J. de St. Germain, T. C. Henderson, and J. C. Owen. Constructing high-precision geometric models from sensed position data. In *Proc. ARPA Image Understanding Workshop*, February 1996.

[Traband *et al.*, 1996] M. T. Traband, F. W. Tillotson, and J. D. Martin. Reverse and re-engineering in the DOD organic maintenance community: Current status and future direction. Technical Report 96-060, Applied Research Laboratory, The Pennsylvania State University, February 1996.

Photorealistic Scene Reconstruction by Voxel Coloring

Steven M. Seitz **Charles R. Dyer**[*]
Department of Computer Science
University of Wisconsin
Madison, WI 53706
E-MAIL {seitz|dyer}@cs.wisc.edu
HOMEPAGE http://www.cs.wisc.edu/~{seitz|dyer}

Abstract

A novel scene reconstruction technique is presented, different from previous approaches in its ability to cope with large changes in visibility and its modeling of intrinsic scene color and texture information. The method avoids image correspondence problems by working in a discretized scene space whose voxels are traversed in a fixed visibility ordering. This strategy takes full account of occlusions and allows the input cameras to be far apart and widely distributed about the environment. The algorithm identifies a special set of invariant voxels which together form a spatial and photometric reconstruction of the scene, fully consistent with the input images.

1 Introduction

We consider the problem of acquiring photorealistic 3D models of real environments from widely distributed viewpoints. This problem has sparked recent interest in the computer vision community [Kanade *et al.*, 1995, Moezzi *et al.*, 1996, Beardsley *et al.*, 1996, Leymarie *et al.*, 1996] as a result of new applications in telepresence, virtual walkthroughs, automatic 3D model construction, and other problems that require realistic textured object models.

We use the term *photorealism* to refer to 3D reconstructions of real scenes whose reprojections contain sufficient color and texture information to accurately reproduce images of the scene from a wide range of target viewpoints. To ensure accurate reprojections, the input images should be representative, i.e., distributed throughout the target range of viewpoints. Accordingly, we propose two criteria that a photorealistic reconstruction technique should satisfy:

- Photo Integrity: The reprojected model should accurately reproduce the color and texture of the input images
- Broad Coverage: The input images should be widely distributed throughout the environment, enabling a wide coverage of scene surfaces

Instead of using existing stereo and structure-from-motion methods to solve this problem, we choose to approach it from first principles. We are motivated by the fact that current reconstruction techniques were not designed with these objectives in mind and, as we will argue, do not fully meet these requirements. Driven by the belief that photo integrity has more to do with color than shape, we formulate a *color reconstruction* problem, in which the goal is an assignment of colors (radiances) to points in an (unknown) approximately Lambertian scene. It is shown that certain points have an invariant coloring, constant across all possible interpretations of the scene, consistent with the input images. This leads to a volumetric *voxel coloring* method that labels the invariant scene

[*] The support of the Defense Advanced Research Projects Agency, and the National Science Foundation under Grant No. IRI-9530985 is gratefully acknowledged.

voxels based on their projected correlation with the input images. By traversing the voxels in a special order it is possible to fully account for occlusions—a major advantage of this scene-based approach. The result is a complete 3D scene reconstruction, built to maximize photo integrity.

The photorealistic scene reconstruction problem, as presently formulated, raises a number of unique challenges that push the limits of existing techniques. First, the reconstruction must be dense and sufficiently accurate to reproduce the original images. This requirement poses a problem for feature-based reconstruction methods, which produce relatively sparse reconstructions. Although sparse reconstructions can be augmented by fitting surfaces (e.g., [Beardsley *et al.*, 1996]), the triangulation techniques currently used cannot easily cope with discontinuities and, more importantly, are not image driven. Consequently, surfaces derived from sparse reconstructions may only agree with the input images at points where image features were detected.

Contour-based methods (e.g., [Cipolla and Blake, 1992, Szeliski, 1993, Seales and Faugeras, 1995]) are attractive in their ability to cope with changes in visibility, but do not produce sufficiently accurate depth-maps due to problems with concavities and lack of parallax information. A purely contour-based reconstruction can be texture-mapped, as in [Moezzi *et al.*, 1996], but not in a way that ensures projected consistency with all of the input images, due to the aforementioned problems. In addition, contour-based methods require occluding contours to be isolated; a difficult segmentation problem avoided by voxel coloring.

The second objective requires that the input views be scattered over a wide area and therefore exhibit large scale changes in visibility (i.e., occlusions, changing field of view). While some stereo methods can cope with limited occlusions, visibility changes of much greater magnitude appear to be beyond the state of the art. In addition, the views may be far apart, making the correspondence problem extremely difficult. Existing stereo-based approaches to this problem [Kanade *et al.*, 1995] match nearby images two or three at a time to ameliorate visibility problems. This approach, however, does not fully integrate the image information and introduces new complications, such as how to merge the partial reconstructions.

The voxel coloring algorithm presented here works by discretizing scene space into a set of voxels that are traversed and colored in a special order. In this respect, the method is similar to Collins' Space-Sweep approach [Collins, 1996], which performs an analogous scene traversal. However, the Space-Sweep algorithm does not provide a solution to the occlusion problem, a primary contribution of this paper. Katayama et al. [Katayama *et al.*, 1995] described a related method in which images are matched by detecting lines through slices of an epipolar volume, noting that occlusions could be explained by labeling lines in order of increasing slope. This ordering is consistent with our results, following from the derivations in Section 2. However, their algorithm used a reference image, thereby ignoring points that are occluded in the reference image but visible elsewhere. Also, their line detection strategy requires that the views all lie on a straight line, a significant limitation. An image-space visibility ordering was described by McMillan and Bishop [McMillan and Bishop, 1995], but was used only for image rendering, not correspondence computation or scene reconstruction. Also related are recently developed panoramic stereo [McMillan and Bishop, 1995, Kang and Szeliski, 1996] algorithms that avoid field of view problems by matching 360 degree panoramic views directly. Panoramic reconstructions can also be achieved using our approach, but without the need to first build panoramic images (see Figures 1(b) and 4).

The remainder of the paper is organized as follows. Section 2 formulates and solves the voxel coloring problem, and describes its relationship to shape reconstruction. Section 3 presents an efficient algorithm for computing the voxel coloring from a set of images. Section 4 describes some experiments on real and synthetic image sequences that demonstrate how the method performs.

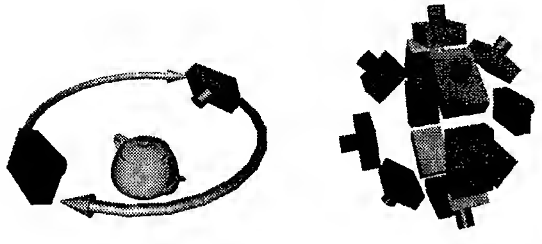


Figure 1: Two camera geometries that satisfy the ordinal visibility constraint.

2 Voxel Coloring

This section describes the voxel coloring problem in detail. The main results require a visibility property that constrains the camera placement relative to the scene, but still permits the input cameras to be spread widely throughout the scene. The visibility property defines a fixed occlusion ordering, enabling scene reconstruction with a single pass through the voxels in the scene.

We assume that the scene is entirely composed of rigid Lambertian surfaces under fixed illumination. Under these conditions, the radiance at each point is isotropic and can therefore be described by a scalar value which we call *color*. We also use the term color to refer to the irradiance of an image pixel. The term's meaning should be clear by context.

2.1 Notation

A 3D scene $\mathcal{S}$ is represented as a finite[1] set of opaque voxels (volume elements), each of which occupies a finite and homogeneous scene volume and has a fixed color. We denote the set of all voxels with the symbol $\mathcal{V}$. An image is specified by the set $\mathcal{I}$ of all its pixels. For now, assume that pixels are infinitesimally small.

Given an image pixel p and scene $\mathcal{S}$, we refer to the voxel $V \in \mathcal{S}$ that is visible and projects to p by $V = \mathcal{S}(p)$. The color of an image pixel $p \in \mathcal{I}$ is given by $color(p, \mathcal{I})$ and of a voxel V by $color(V, \mathcal{S})$. A scene $\mathcal{S}$ is said to be *complete* with respect to a set of images if, for every image $\mathcal{I}$ and every pixel $p \in \mathcal{I}$, there exists a voxel $V \in \mathcal{S}$ such that $V = \mathcal{S}(p)$. A complete scene is said to be *consistent* with a set of images if, for every image $\mathcal{I}$ and every pixel $p \in \mathcal{I}$,

$$color(p, \mathcal{I}) = color(\mathcal{S}(p), \mathcal{S}) \qquad (1)$$

2.2 Camera Geometry

A pinhole perspective projection model is assumed, although the main results use a visibility assumption that applies equally to other camera models such as orthographic and aperture-based models. We require that the viewpoints (camera positions) are distributed so that ordinal visibility relations between scene points are preserved. That is, if scene point P occludes Q in one image, Q cannot occlude P in any other image. This is accomplished by ensuring that all viewpoints are "on the same side" of the object. For instance, suppose the viewpoints are distributed on a single plane, as shown in Figure 1(a). For every such viewpoint, the relative visibility of any two points depends entirely on which point is closer to the plane. Because the visibility order is fixed for every viewpoint, we say that this range of viewpoints preserves ordinal visibility.

Planarity, however, is not required; the ordinal visibility constraint is satisfied for a relatively wide range of viewpoints, allowing significant flexibility in the image acquisition process. Observe that the constraint is violated only when there exist two scene points P and Q such that P occludes Q in one view while Q occludes P in another. This condition implies that P and Q lie on the line segment between the two camera centers. Therefore, a sufficient condition for the ordinal visibility constraint to be satisfied is that **no scene point be contained within the convex hull $\mathcal{C}$ of the camera centers.** For convenience, $\mathcal{C}$ will be referred to as the *camera volume*. We use the notation $dist(V, \mathcal{C})$ to denote the distance of a voxel V to the camera volume. Figure 1 shows two useful camera geometries that satisfy this constraint, one a downward facing camera moved 360 degrees around an object, and the other outward facing cameras on a sphere.

[1] It is assumed that the visible scene is spatially bounded.

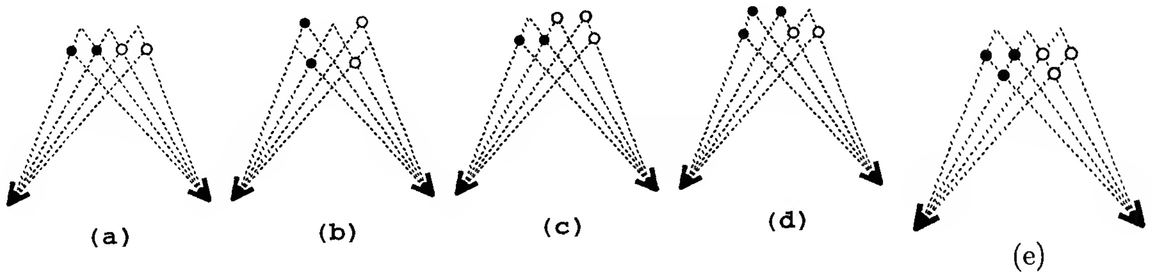


Figure 2: (a-d) Four scenes that are indistinguishable from these two viewpoints. Shape ambiguity: scenes (a) and (b) have no points in common—no hard points exist. Color ambiguity: (c) and (d) share a point that has a different color assignment in the two scenes. (e) The voxel coloring produced from the two images in (a-d). These six points have the same color in every consistent scene that contains them.

2.3 Color Invariance

It is well known that a set of images can be consistent with more than one rigid scene. Determining a scene's spatial occupancy is therefore an ill-posed task because a voxel contained in one consistent scene may not be contained in another (Figure 2(a,b)). Alternatively, a voxel may be part of two consistent scenes, but have different colors in each (Figure 2(c,d)).

Given a multiplicity of solutions to the reconstruction problem, the only way to recover intrinsic scene information is through *invariants*— properties that are satisfied by every consistent scene. For instance, consider the set of voxels that are present in every consistent scene. Laurentini [Laurentini, 1995] described how these invariants, called *hard points*, could be recovered by volume intersection from binary images. Hard points are useful in that they provide absolute information about the true scene. However, such points can be difficult to come by; some images may yield none (e.g., Figure 2). In this section we describe a more frequently occurring type of invariant relating to color rather than shape.

> A voxel V is a **color invariant** with respect to a set of images if, for every pair of scenes $\mathcal{S}$ and $\mathcal{S}'$ consistent with the images, $V \in \mathcal{S}, \mathcal{S}'$ implies $color(V, \mathcal{S}) = color(V, \mathcal{S}')$

Unlike shape invariance, color invariance does not require that a point be present in every consistent scene. As a result, color invariants tend to be more common than hard points. In particular, any set of images satisfying the ordinal visibility constraint yields enough color invariants to form a complete scene reconstruction, as will be shown.

Let $\mathcal{I}_1, \ldots, \mathcal{I}_m$ be a set of images. For a given image point $p \in \mathcal{I}_j$ define V_p to be the voxel in $\{\mathcal{S}(p) \mid \mathcal{S}\ consistent\}$ that is closest to the camera volume. We claim that V_p is a color invariant. To establish this, observe that $V_p \in \mathcal{S}$ implies $V_p = \mathcal{S}(p)$, for if $V_p \neq \mathcal{S}(p)$, $\mathcal{S}(p)$ must be closer to the camera volume, which is impossible by the construction of V_p. It then follows from Eq. (1) that V_p has the same color in every consistent scene; V_p is a color invariant.

> The **voxel coloring** of an image set $\mathcal{I}_1, \ldots, \mathcal{I}_m$ is defined to be:
> $\overline{\mathcal{S}} = \{V_p \mid p \in \mathcal{I}_i,\ 1 \leq i \leq m\}$

Figure 2(e) shows the voxel coloring resulting from a pair of views. These six points have a unique color interpretation, constant in every consistent scene. They also comprise the closest consistent scene to the cameras in the following sense—every point in each consistent scene is either included in the voxel coloring or is fully occluded by points in the voxel coloring. An interesting consequence of this closeness bias is that neighboring image pixels of the same color produce cusps in the voxel coloring, i.e., protrusions toward the camera volume. This phenomenon is clearly shown in Figure 2(e) where the white and black points form two separate cusps. Also, observe that the voxel coloring is not a minimal reconstruction; removing the two closest points in Figure 2(e) still leaves a consistent scene.

2.4 Computing the Voxel Coloring

In this section we describe how to compute the voxel coloring from a set of images. In addition it will be shown that the set of voxels contained in a voxel coloring form a scene reconstruction that is consistent with the input images.

The voxel coloring is computed one voxel at a time in an order that ensures agreement with the images at each step, guaranteeing that all reconstructed voxels satisfy Eq. (1). To demonstrate that voxel colorings form consistent scenes, we also have to show that they are complete, i.e., they account for every image pixel as defined in Section 2.1.

In order to make sure that the construction is incrementally consistent, i.e., agrees with the images at each step, we need to introduce a weaker form of consistency that applies to incomplete voxel sets. Accordingly, we say that a set of points with color assignments is *voxel-consistent* if its projection agrees fully with the subset of every input image that it overlaps. More formally, a set $\mathcal{S}$ is said to be voxel-consistent with images $\mathcal{I}_1, \ldots, \mathcal{I}_m$ if for every voxel $V \in \mathcal{S}$ and image pixels $p \in \mathcal{I}_i$ and $q \in \mathcal{I}_j$, $V = \mathcal{S}(p) = \mathcal{S}(q)$ implies $color(p, \mathcal{I}_i) = color(q, \mathcal{I}_j)$. For notational convenience, define $\mathcal{S}_V$ to be the set of all voxels in $\mathcal{S}$ that are closer than V to the camera volume. Scene consistency and voxel consistency are related by the following properties:

1. If $\mathcal{S}$ is a consistent scene then $\{V\} \cup \mathcal{S}_V$ is a voxel-consistent set for every $V \in \mathcal{S}$.

2. Suppose $\mathcal{S}$ is complete and, for each point $V \in \mathcal{S}$, $V \cup \mathcal{S}_V$ is voxel-consistent. Then $\mathcal{S}$ is a consistent scene.

A consistent scene may be created using the second property by incrementally moving further from the camera volume and adding voxels to the current set that maintain voxel-consistency. To formalize this idea, we define the following partition of 3D space into voxel layers of uniform distance from the camera volume:

$$\mathcal{V}_\mathcal{C}^d = \{V \mid dist(V, \mathcal{C}) = d\} \qquad (2)$$

$$\mathcal{V} = \bigcup_{i=1}^{r} \mathcal{V}_\mathcal{C}^{d_i} \qquad (3)$$

where $d_1, \ldots, d_r$ is an increasing sequence of numbers.

The voxel coloring is computed inductively as follows:

$$\begin{aligned} \mathcal{SP}_1 &= \{V \mid V \in \mathcal{V}_{d_1}, \{V\} \text{ voxel-consistent}\} \\ \mathcal{SP}_k &= \{V \mid V \in \mathcal{V}_{d_k}; \\ &\qquad \{V\} \cup \mathcal{SP}_{k-1} \text{ voxel-consistent}\} \\ \mathcal{SP} &= \{V \mid V = \mathcal{SP}_r(p) \text{ for some pixel } p\} \end{aligned}$$

We claim $\mathcal{SP} = \overline{\mathcal{S}}$. To prove this, first define $\overline{\mathcal{S}}_i = \{V \mid V \in \overline{\mathcal{S}}, dist(V, \mathcal{C}) \leq d_i\}$. $\overline{\mathcal{S}}_1 \subseteq \mathcal{SP}_1$ by the first consistency property. Inductively, assume that $\overline{\mathcal{S}}_{k-1} \subseteq \mathcal{SP}_{k-1}$ and let $V \in \overline{\mathcal{S}}_k$. By the first consistency property, $\{V\} \cup \overline{\mathcal{S}}_{k-1}$ is voxel-consistent, implying that $\{V\} \cup \mathcal{SP}_{k-1}$ is also voxel-consistent, because the second set includes the first and $\mathcal{SP}_{k-1}$ is itself voxel-consistent. It follows that $\overline{\mathcal{S}} \subseteq \mathcal{SP}_r$. Note also that $\mathcal{SP}_r$ is complete, since one of its subsets is complete, and hence consistent by the second consistency property. $\mathcal{SP}$ contains all the voxels in $\mathcal{SP}_r$ that are visible in any image, and is therefore consistent as well. Therefore $\mathcal{SP}$ is a consistent scene such that for each pixel p, $\mathcal{SP}(p)$ is at least as close to $\mathcal{C}$ as $\overline{\mathcal{S}}(p)$. Hence $\mathcal{SP} = \overline{\mathcal{S}}$. □

In summary, the following properties of voxel colorings have been shown:

- $\overline{\mathcal{S}}$ is a consistent scene
- Every voxel in $\overline{\mathcal{S}}$ is a color invariant
- $\overline{\mathcal{S}}$ is directly computable from any set of images satisfying the ordinal visibility constraint

3 Reconstruction by Voxel Coloring

In this section we present a voxel coloring algorithm for reconstructing a scene from a set of calibrated images. The algorithm closely follows the voxel coloring construction outlined in Section 2, adapted to account for image quantization and noise. As before, it is assumed that 3D space has been partitioned into a series of voxel layers $\mathcal{V}_{\mathcal{C}}^{d_1}, \ldots, \mathcal{V}_{\mathcal{C}}^{d_r}$ increasing in distance from the camera volume. The images $\mathcal{I}_1, \ldots, \mathcal{I}_m$ are assumed to be quantized into finite nonoverlapping pixels. The cameras are assumed to satisfy the ordinal visibility constraint, i.e., no scene point lies within the camera volume.

If a voxel V is not fully occluded in image $\mathcal{I}_j$, its projection will overlap a nonempty set of image pixels, π_j. Without noise or quantization effects, a consistent voxel should project to a set of pixels with equal color values. In the presence of these effects, we evaluate the correlation of the pixel colors to measure the likelihood of voxel consistency. Let s be the standard deviation and n the cardinality of $\bigcup_{j=1}^{m} \pi_j$.

Suppose the sensor error (accuracy of irradiance measurement) is approximately normally distributed with standard deviation σ_0. If σ_0 is unknown, it can be estimated by imaging a homogeneous surface and computing the standard deviation of image pixels. The consistency of a voxel can be estimated using the following likelihood ratio test, distributed as χ^2:

$$\lambda_V = \frac{(n-1)s}{\sigma_0}$$

3.1 Voxel Coloring Algorithm

The algorithm is as follows:

```
$\mathcal{S} = \emptyset$
for $i = 1, \ldots, r$ do
    for every $V \in \mathcal{V}_{\mathcal{C}}^{d_i}$ do
        project to $\mathcal{I}_1, \ldots, \mathcal{I}_m$, compute $\lambda_V$
        if $\lambda_V < thresh$ then $\mathcal{S} = \mathcal{S} \cup \{V\}$
```

The threshold, *thresh*, corresponds to the maximum allowable correlation error. An overly conservative (small) value of *thresh* results in an accurate but incomplete reconstruction. On the other hand, a large threshold yields a more complete reconstruction, but one that includes some erroneous voxels. In practice, *thresh* should be chosen according to the desired characteristics of the reconstructed model, in terms of accuracy vs. completeness.

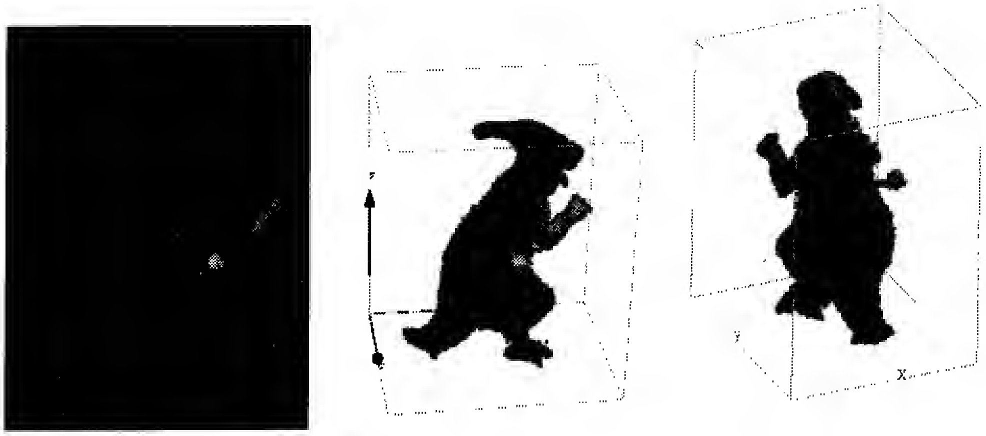
The problem of detecting occlusions is greatly simplified by the scene traversal ordering used in the algorithm; the order is such that if V occludes V' then V is visited before V'. Therefore, occlusions can be detected by using a one-bit Z-buffer for each image. The Z-buffer is initialized to 0. When a voxel V is processed, π_i is the set of pixels that overlap V's projection in $\mathcal{I}_i$ and have Z-buffer values of 0. Once λ_V is calculated, these pixels are then marked with Z-buffer values of 1.

3.2 Discussion

The algorithm visits each voxel exactly once and projects it into every image. Therefore, the time complexity of voxel coloring is: $O(voxels * images)$. To determine the space complexity, observe that evaluating one voxel does not require access to or comparison with other voxels. Consequently, voxels need not be stored during the algorithm; the voxels making up the voxel coloring will simply be output one at a time. Only the images and one-bit Z-buffers need to be stored. The fact that the complexity of voxel coloring is linear in the number of images is essential in that it enables large sets of images to be processed at once.

The algorithm is unusual in that it does not perform any window-based image matching in the reconstruction process. Correspondences are found implicitly during the course of scene traversal. A disadvantage of this searchless strategy is that it requires very precise camera calibration to achieve the triangulation accuracy of existing stereo methods. Accuracy also depends on the voxel resolution.

Importantly, the approach reconstructs only one of the potentially numerous scenes consistent with the input images. Consequently, it is susceptible to aperture problems caused by image regions of near-uniform color. These regions will produce cusps in the reconstruction (see Figure 2(e)) since voxel coloring seeks the re-



(a) (b) (c)

Figure 3: Reconstruction of a dinosaur toy. (a) One of 21 input images taken from slightly above the toy while it was rotated 360°. (b-c) Two views rendered from the reconstruction.

construction closest to the camera volume. This is a bias, just like smoothness is a bias in stereo methods, but one that guarantees a consistent reconstruction even with severe occlusions.

4 Experimental Results

The first experiment involved reconstructing a dinosaur toy from 21 views spanning a 360 degree rotation of the toy. Figure 3 shows the voxel coloring computed. To facilitate reconstruction, we used a black background and eliminated most of the background points by thresholding the images. While background subtraction is not strictly necessary, leaving this step out results in background-colored voxels scattered around the edges of the scene volume. The threshold may be chosen conservatively since removing most of the background pixels is sufficient to eliminate this background scattering effect. Figure 3(b) shows the reconstruction from approximately the same viewpoint as (a) to demonstrate the photo integrity of the reconstruction. Figure 3(c) shows another view of the reconstructed model. Note that fine details such as the wind-up rod and hand shape were accurately reconstructed. The reconstruction contained 32,244 voxels and took 45 seconds to compute.

A second experiment involved reconstructing a synthetic room from views *inside* the room. The room interior was highly concave, making accurate reconstruction by volume intersection or other contour-based methods impractical. Figure 4 compares the original and reconstructed models from new viewpoints. New views were generated from the room interior quite accurately, as shown in (a), although some details were lost. For instance, the reconstructed walls were not perfectly planar. This point drift effect is most noticeable in regions where the texture is locally homogeneous, indicating that texture information is important for accurate reconstruction. The reconstruction contained 52,670 voxels and took 95 seconds to compute.

5 Concluding Remarks

This paper presented a new scene reconstruction technique that incorporates intrinsic color and texture information for the acquisition of photorealistic scene models. Unlike existing stereo and structure-from-motion techniques, the method *guarantees* that a consistent reconstruction is found, even under severe visibility changes, subject to a weak constraint on the camera geometry. A second contribution was the constructive proof of the existence of a set of color invariants. These points are useful in two ways: first, they provide information that is intrinsic, i.e., constant across all possible consistent scenes. Second, together they constitute a volumetric spatial reconstruction of the scene whose projections exactly match the input images.

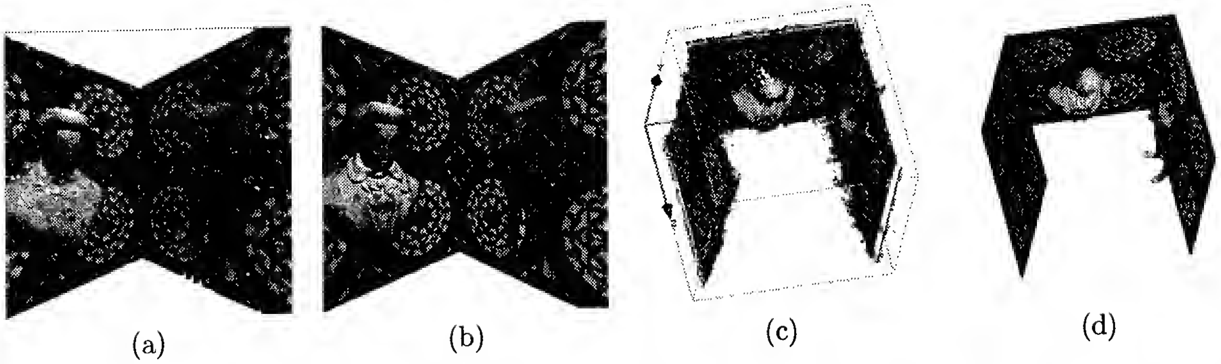


Figure 4: Reconstruction of a synthetic room scene. (a) The voxel coloring. (b) The original model from a new viewpoint. (c) and (d) show the reconstruction and original model, respectively, from a new viewpoint outside the room.

References

[Beardsley *et al.*, 1996] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. European Conf. on Computer Vision, Vol. 2*, pages 683–695, 1996.

[Cipolla and Blake, 1992] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *Int. J. Computer Vision*, 9(2):83–112, 1992.

[Collins, 1996] R. Collins. A space-sweep approach to true multi-image matching. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 358–363, 1996.

[Kanade *et al.*, 1995] T. Kanade, P. Narayanan, and P. Rander. Virtualized reality: Concepts and early results. In *Proc. IEEE Workshop on Representation of Visual Scenes*, pages 69–76, 1995.

[Kang and Szeliski, 1996] S. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 364–370, 1996.

[Katayama *et al.*, 1995] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura. A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. In *Proc. SPIE Vol. 2409A*, pages 21–30, 1995.

[Laurentini, 1995] A. Laurentini. How far 3D shapes can be understood from 2D silhouettes. *IEEE Trans. Pattern Analysis and Machine Intell.*, 17(2):188–195, 1995.

[Leymarie *et al.*, 1996] F. Leymarie, A. de la Fortelle, J. Koenderink, A. Kappers, M. Stavridi, B. van Ginneken, S. Muller, S. Krake, O. Faugeras, L. Robert, C. Gauclin, S. Laveau, and C. Zeller. Realise: Reconstruction of reality from image sequences. In *Proc. Int. Conf. Image Processing, Vol. 3*, pages 651–654, 1996.

[McMillan and Bishop, 1995] Leonard McMillan and Gary Bishop. Plenoptic modeling. In *Proc. SIGGRAPH 95*, pages 39–46, 1995.

[Moezzi *et al.*, 1996] S. Moezzi, A. Katkere, D. Kuramura, and R. Jain. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, 1996.

[Seales and Faugeras, 1995] W. Seales and O. Faugeras. Building three-dimensional object models from image sequences. *CVGIP: Image Understanding*, 3(61):308–324, 1995.

[Szeliski, 1993] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 1(58):23–32, 1993.

Edge-aligning Surface Fitting Using Triangular B-Splines

Song Han and Gérard Medioni

Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, CA 90089
han, medioni@iris.usc.edu

Abstract

We propose a new surface fitting scheme called "winged B-snakes" for reconstructing smooth surfaces and concurrently preserving discontinuity edges. The input data can be noisy, sparse, scattered 3D data on a surface with unknown crease edges. First, we use a vector voting method to produce three potential fields for surfaces, edges, and junctions. The potential fields are stored in three volumetric grids, giving each voxel the *probability* of being a surface point, an edge point, and a junction point. Then we drop a deformable surface in the potential fields for evolution. The surface representation is the latest triangular B-spline model. By performing energy minimization in the potential fields, each active edge slides to align with discontinuity edges, and its wing patches flap to fit to the surface data. Finally, a smooth C^1 surface preserving discontinuity edges and junctions is constructed.

1 The Problem

Geometric models play an important role in graphics, CAD/CAM, visualization, animation, multimedia, computer vision, and many other fields [1]. Geometric models can be designed, stored, and processed as volumetric primitives or surface patches. Complex models can be built more efficiently by reconstructing them from an already existing objects or clay-models than by designing them from scratch using a CAD software. Surface reconstruction is the process of building surface models from sampled data.

The sampled data have different forms: one or a few intensity images, slices extracted from CT/MRI data, dense regularly-gridded range images scanned by a laser beam, sparse coordinates measured by a probe, etc. To be more general, we use noisy sparse scattered 3D data [2]. Nevertheless, our proposed new scheme is also applicable to dense range images but some intermediate stages can be made more efficient if simpler methods are used for dense data on a regular grid.

In recent years, deformable surfaces based on regularization theory seem to have provided a systematic approach to handling the uncertainty and ill-posedness in the noisy sparse scattered data, and thus have become the focus of surface reconstruction research [15]. Even in CAD/CAGD, the deformable-surface-based variational design has become a popular scheme [4]-[12]. The spirit in deformable surface is that besides the external fitting energy pulling the surface toward the data points, a smoothness constraint as an internal energy is introduced to regularize the surface itself so that the reconstructed surface will not go wild in very noisy regions and can still be defined over gaps where there are no measured data.

However, when the data contains multiple objects, or an object has discontinuity edges on its surface, such overall regularization may fail, hence an oversmoothed surface is obtained, and the discontinuity features may be lost. Also, there exist overshoots/undershoots and ringings (Gibbs effects) around the discontinuities. So we wish to be given the discontinuity edges before we do surface reconstruction. However, automatic edge detection, linking, localization and classification are hard problems, and to find unknown edges, we would require a faithful surface representation for the raw data. Due to this circular dependence between surface reconstruction and edge detection, researchers have realized that they are two coupled processes and cannot be well solved separately. Based on this observation, we propose a novel approach to surface reconstruction coupled with edge detection. We do not suppose known discontinuities on the surface. Our approach will perform surface reconstruction and discontinuity detection concurrently.

* This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by Topographic Engineering Center of the U.S. Army.

Guy and Medioni [3] have implemented an automatic algorithm for discontinuity detection and surface fitting, which produces dense triangular meshes and discontinuity curves automatically from scattered data. In this paper, we use their approach to infer potential information for surfaces and discontinuities, but the surface representation is not the planar triangular meshes. Instead, we use smooth triangular B-spline surfaces.

Triangular B-splines [4]-[8] overcome the shortcomings of tensor-product (TP) B-splines but retain the good properties of automatic continuity, convex hull, etc. TP-splines have a rectangular topology and thus require tedious trimming techniques to handle pole artifacts and irregular boundaries; also, to obtain C^1 continuity, the TP B-splines should be quadratic in terms of both parameters, so the total degree is four (quartic); whereas for triangular splines, a total degree of as low as two (quadratic) can maintain the C^1 continuity; duplicate knots in TP splines will produce a discontinuity curve across the whole surface, while triangular splines can produce a local discontinuity edge. Similarly, triangular splines allow for local subdivision for refinement.

Our approach starts with a grouping stage to infer dense potential information from the sparse data. In the second stage, a deformable triangular surface coupled with active edges is dropped into the potential fields. We call the model "winged B-snakes". After adjusting the control points by a few iterations of energy minimization, the surface (wings) flap to fit the data, and the edges (snakes) slide to align with the actual edges in the data. Then in the third stage, values and derivatives along each edge are checked, so that discontinuities can be detected and preserved in constructing the surface. The surface can also be extended to rational forms and the weights are adjusted for surface fine-tuning and fairing. An introduction to triangular B-splines is given in Section 2. In Section 3, we explain the principles of the winged B-snakes, with some experimental results given in Section 4. Further topics are discussed in Section 4 and we conclude the paper in Section 5.

2 The Representation

Triangular B-splines are defined over 2D domain triangulations, and C^{k-1} continuity can be achieved by k-th degree polynomials defined over the domain triangles. In our work, we consider C^1 surfaces using quadratic polynomials (k=2). Given an arbitrary triangulation of the 2D domain, two additional points are added near each vertex of the triangle [v_i, v_j, v_k] to provide nine knots for each triangle (two adjacent triangles share six knots). Then from the nine knots, five knots (including the original vertices v_i, v_j, v_k, plus two additional knots) are chosen to form a knot set. Six different knot sets are chosen; and over each set, a basis function is defined.

A basis function B over the 5-knot set K can be defined recursively as follows:

$$B(u|K) = a_0 B(u|K\backslash v_0) + a_1 B(u|K\backslash v_1) + a_2 B(u|K\backslash v_2)$$

where u is any point in the 2-D domain, a_0, a_1, a_2 are related to the barycentric coordinates of u w.r.t. any three knots v_0,v_1,v_2; $K\backslash v_i$ is the 5-knot set K minus v_i, i.e., a 4-knot set. As the $K\backslash v_i$ reduces to only 3 knots, a zeroth-degree B-spline basis function is obtained, which is a flat unit-height triangle:

$$B(u|v_0, v_1, v_2) = \begin{cases} 1 & \text{if} \quad u \in [v_0, v_1, v_2) \\ 0 & \text{otherwise} \end{cases}$$

It can be shown that three collinear knots will result in a crease edge while four collinear knots will produce a step edge. If the nine knots are pulled apart and no collinear knots exist at all, each B-spline basis function automatically becomes C^1. Linear combination of the six basis functions give rise to a local patch, and over the whole domain triangulation (T triangles) a surface is obtained:

$$X = \sum_{t=0}^{T-1} \sum_{b=0}^{5} c_{t,b} \cdot B_{t,b}(u)$$

where X is a point (x,y,z) in 3D space, $c_{t,b}$'s are the scaling factors to the basis functions acting as control points (adjacent triangles may share 3D control points to guarantee C^0, or share their xy components but not z component to allow a step edge). If a weight $w_{t,b}$ is associated to each control point, the above triangular B-spline surface is extended to triangular NURBS surface:

$$X = \frac{\sum_{t=0}^{T-1} \sum_{b=0}^{5} c_{t,b} w_{t,b} \cdot B_{t,b}(u)}{\sum_{t=0}^{T-1} \sum_{b=0}^{5} w_{t,b} \cdot B_{t,b}(u)}$$

Note that if all the weights are equal, the triangular NURBS specialize to triangular B-splines; furthermore, if the three pulled-apart knots collapse to be duplicate at each vertex, then the overlap/blend effect among adjacent basis functions disappear, and thus triangular B-splines degenerate to triangular Bezier-splines, and the automatic continuity property is lost.

If the above domain is a triangulation of the surface of a unit sphere, a closed surface is defined. The only difference is that the summation of three spherical barycentric coordinates are not 1 anymore (it is usually greater than 1), and that three or four knots on a same great circle will produce a discontinuity. Such a spherical representation is very useful in geographical and other applications. Since the splines are defined over arbitrary triangulations of the sphere, there are no pole artifact as in the TP splines, and the continuity is automatically guaranteed. The simplest case is the tetrahedron tessellation of the unit sphere, in which as few as four triangles suffice to exactly model a sphere. By contrast, to cover a sphere with rectangles, at least six rectangles (the faces of a cube) are required to obtain the continuity, and a set of constraint equations have to be maintained.

3 Surface Reconstruction

3.1 Inferring Potential Fields

The goal of the first stage is to infer dense potential measures from the scattered data, as described in the flow chart in Figure 1. The input can be points, segments, or patches. In our work, we only use points. The details about the method can be found in Guy and Medioni's paper [3].

The idea is to locally enforce the general constraints, which are *co-surfacity*, *proximity*, and *constancy of curvature*. These constraints are encoded into a 3D vector mask. Such a mask, when aligned with an input data site, associates a preferred direction and strength to every voxel in a large volume of space around the input site. By aligning the field with each input site, we produce, at each voxel location, a collection of vector votes. This voting information is then compressed into the second order moments by the covariance matrix, graphically represented by an ellipsoid, or equivalently, by three eigen-vectors. The eigen-values l_{max}, l_{mid}, l_{min} are interpreted as three saliency measures for surfaces, edges and junctions, and the eigen-vectors are used to estimate the surface normals.

In more details, l_{max}- l_{mid} is used as the saliency of a surface passing through a location, since if l_{max}- l_{mid} is large, l_{max} will be large and l_{mid} is small, also l_{min} is small (since $l_{min}< l_{mid}$). Thus, there is only one strong vote group here, i.e., the consistency of votes at this location is high. In other words, the probability of a real surface passing through this location is high. Similarly, l_{mid} - l_{min} is used as an edge saliency measure, and l_{min} is used as junction saliency measure.

Note that this voting methodology imposes no restriction on the number of objects, genus (topology), number of discontinuities, and the algorithm is non-iterative and efficient. By negating the above three saliency measures, we obtain the potential fields for surfaces, edges and junctions. The minimum potential locations of the three potential fields indicate the existence of surfaces, edges and junctions, respectively.

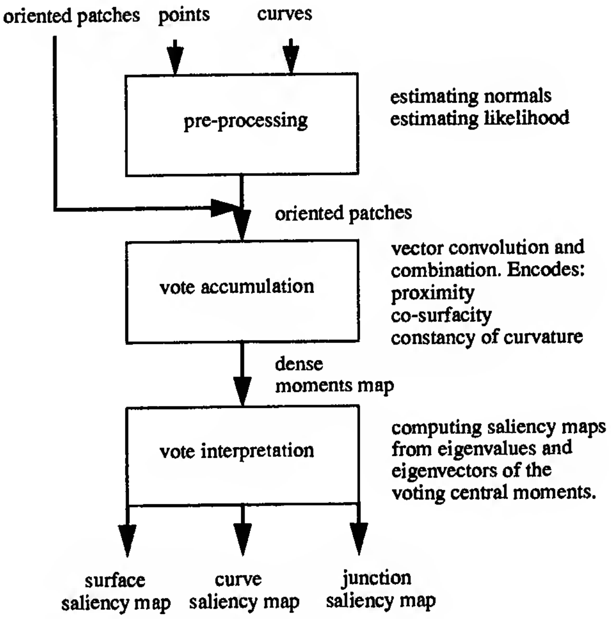


Figure 1. Flow chart of the grouping stage

The above voting procedures are performed for each voxel in a 3D grid, and they can work for any number of surfaces of arbitrary topology. They even work for non-manifolds. The voting complexity is $O(n^3k)$ in general, where n is the side size of the volume grid, and k is the number of data points.

3.2 Edge-aligning Surface Deformation

From the surface potential field, a triangular mesh can be traced out by the "marching cube" algorithm [25]. Different from the standard marching cube algorithm, the surface to be traced out is now the minimum-potential surface or zero-derivative surface of the potential field, instead of the iso-surface of the potential values. Similarly, the discontinuity curves and junctions can be traced out by marching methods also. However, the mesh is quite dense with each facet being a triangle; the edges and junctions are traced out from separate fields and are not integrated into the surfaces.

Our goal is to obtain a sparse curved surface representation, with the discontinuity edges and junctions preserved in the surfaces. We drop a triangular spline surface into the potential fields, and deform the surface to reach minimum energy. Different from previous work, we couple the deformable surface model with the active edge model, so that a fitted surface together with aligned edges will be obtained, and the result gives a compact integrated representation from the three potential fields. Since each surface patch can deform and its edges can slide, we call our model "winged B-snakes".

We define the energy E for the winged B-snakes as follows:

$$E = E_{surface} + E_{edges} + E_{junctions}$$

$$= E_{s\text{-}smooth} + w_1 \cdot E_{s-fit} + w_2 \cdot E_{e-align} + w_3 \cdot E_{j\text{-}align}$$

where w_1, w_2, and w_3 are coefficients to balance the effects of the three types of energy, and are chosen by normalizing the above four terms to be 1 : 20 : 5 : 5 before the start of minimization. They bring the four terms to comparable orders of magnitude, and the minimization results are stable with their variations. We adjust the control points to minimize the total energy.

(1) Surface Smoothness Energy

Although each triangular patch is a quadratic polynomial which is always continuous, we still need the smoothness energy to minimize the mesh roughness. The smoothness energy is defined in terms of the first-order right-hand derivatives:

$$E_{s-smooth} = \sum_{t,X} \iint_{uv} \left(\frac{\partial X^2}{\partial u} + \frac{\partial X^2}{\partial v}\right) dudv$$

where the summation is over all triangles and *(x,y,z)*, and the integration is over the barycentric coordinates (u,v) within a triangle. Since the goal of this stage is mainly surface fitting, we simply use the above membrane energy. The more costly thin-plate energy based on curvature or second-order derivatives is used in the final subtle modification and fairing stage.

(2) Surface Fitting Energy

Inside the surface potential field, the surface patches (wings) flap to reach the local minimum. If the triangle edges cross-over the actual discontinuity edges, the surface energy becomes large; when the edges move to align with the actual edges, the surface energy is reduced. However, we found that the surface energy makes the edges move a little bit, but is not strong enough to pull them to exactly align with the actual discontinuity edges. This is why we shall introduce an explicit edge alignment energy to make the edges become "active" by themselves.

(3) Edge and Junction Alignment Energy

Inside the edge potential field, the triangle edges (snakes) can slide to the local minimum so that the triangle edges may align with the actual discontinuity edges, instead of crossing over them. Also, the triangle vertices can move to the local minimum in the junction potential field, making the vertices align with the inferred junctions. Such alignment significantly reduces the total energy or cooperatively improves the surface fitting precision, and we thus do not have to subdivide the triangles into many tiny ones to obtain a good fitting due to the misaligned edges and junctions as in most previous methods.

3.3 Obtaining C^1 Surfaces with Creases

After the surface patches have been fitted to the data, and the edges and junctions have aligned with actual ones, the detection of discontinuity edges and junctions from the surface is straightforward [8]. We simply need to check the value/derivative differences along the boundary between every pair of adjacent triangles. We normalize the differences by the total area of the pair of triangles. Such locally adaptive thresholding works well. We then check the change of angles between adjacent edge segments along the detected edges, and those vertices with sharp changes are marked as corners, and those with more than two neighbors are marked as junctions. To build the C^1 smooth surface while preserving the edges and junctions, we pull apart the knots in the continuous regions, and allow knots to be duplicate or collinear/co-circular to respect edges/junctions (at first all three knots are duplicate at each vertex). With the new knot configuration, we adjust the control points once more.

3.4 Fine-tuning/Fairing by Adjusting Weights

In the above fitting and alignment procedures, we never adjust the weights. The reasons are that we want to exploit the descriptive power of control points as much as possible; there exist much redundancy in the weights (scaling of weights does not change the shape at all), so leaking information to the weights is not desired. Also, adjusting too many parameters is difficult for the minimization routines. We tested adjusting the weights at the same time as moving the control points, the deformation and alignment slow down from 10 minutes to 30 minutes; more memory is also needed for the Hessian matrix in the minimization routine. Due to the $c_{t,b} \cdot w_{t,b}$ multiplication terms, the minimization becomes non-linear. After some experimental tests, we found that if the initial guess is very good, simultaneous adjustment results in a little improvement (about 1~5% further reduction of the total energy), and that if the initial guess is not good, the residual energy may even be larger! Including knots in the minimization makes the situation even worse, since the knots are buried in the quadratic basis functions. In summary, it seems that adjusting control points and weights simultaneously is not always better, or is not worth the cost, even if it can bring slight improvement.

We use the thin-plate energy as the surface fairness measure, which is based on the second-order derivatives. The energy for fine-tuning and fairing is defined as

$$E = E_{surface-fairness} + c \cdot E_{surface-fit}$$

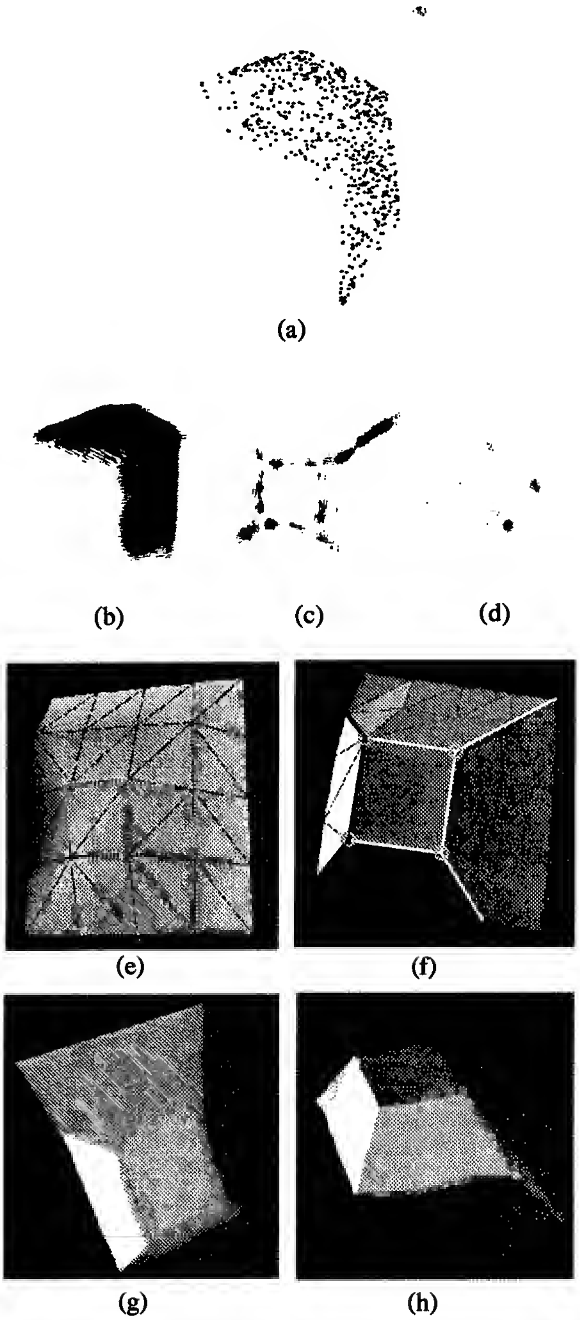
where the coefficient c is chosen such that the two terms are normalized to 1 : 5 before the minimization starts. Since the surface-fitting has been done in stage 3 where the smoothness/fitting ratio is 1 : 20, we can now use smaller ratio (1 : 5). To minimize this energy, only small changes of weights are necessary, and we did not observe negative weights occur in our experiments. Initially all weights are 1's, then later they stay between 0.5 ~ 10. We did not use a negative weight penalty term, such as $\{min(0, w_{t,b})\}^2$ [22]-[24].

4 Experiments

Figure 2(a) is the randomly sampled data (400 points) of a truncated pyramid. (b), (c), (d) display the three volumetric potential fields for surfaces, edges and junctions, respectively (only voxels with potentials below a threshold are displayed). This stage runs for about 10 minutes on a SUN Sparc 10, using three 50x50x50 arrays.

To obtain an initial surface, we regularly split the bounding box of the data points into rectangles and then split each rectangle into two triangles along a diagonal. Which diagonal is chosen depends on which one yields smaller surface fitting energy within this rectangle. Such an initial surface is shown in Figure 2(e) with misaligned edges and junctions.

Due to the model's local control property, a control point only affects a triangle and its surrounding triangles, so it is reasonable to justify that a local minimization may not sacrifice the global optimality. In our implementation, we adjust only a triangle and all its neighboring triangles at a time, and then move to the next triangle. We use the Levenberg-Marquardt algorithm with numerically estimated gradients. Figure 2(f) shows aligned/detected edges and junctions after adjusting the control points. (g) is the C^1 surface with edges and junctions preserved by pulling apart or setting collinear knots then adjusting the control points again. Finally, (h) gives the result after the fine-tuning and fairing by adjusting the weights. We can see that the edges in (i) are sharper than (h), which indicates the effect of adjusting the weights for final-stage subtle improvement. The three stages for reconstruction from the potential fields run for about 10 minutes on an SGI/Indigo.


(a)

(b) (c) (d)

(e) (f)

(g) (h)

Figure 2. Open pyramid

To test the triangular B-splines defined over a unit sphere, we add some samples at the bottom of the pyramid to make it a closed surface, as shown in Figure 3(a). A triangular tessellation of the unit sphere is given in (b) by subdividing and flipping the diagonals on the faces of a unit cube. The initial surface is specified by a sphere enclosing the data set. After the surface fitting and edge alignment, the C^1 result with preserved edges and junctions is shown in (c). Then, (d) gives the final result after fine-tuning and fairing by adjusting the weights. Every triangle is treated equally, and no pole artifact regions exist at all. This demonstrates the major advantage of triangular splines over rectangular tensor-product splines for modeling spherical data. Previous work using rectangular splines had to tolerate the pole degeneracy, or use two or more pieces of surfaces and then glue them together.

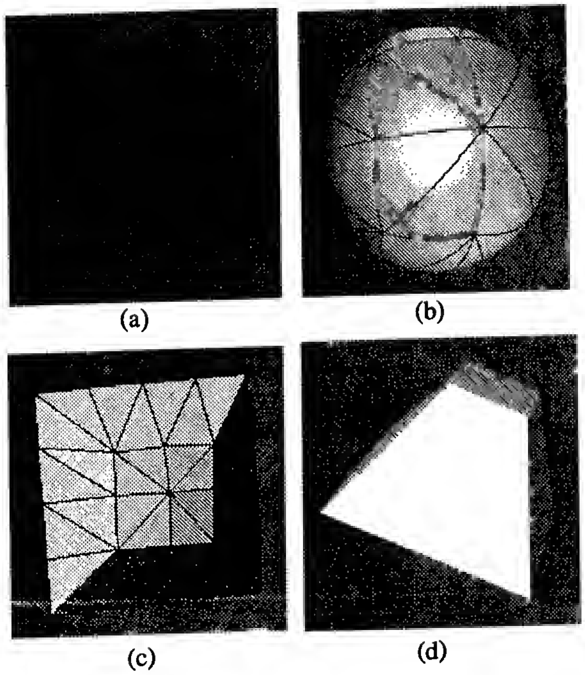


Figure 3. Closed pyramid

Figure 4(a) shows 900 scattered samples of an airplane. We apply a Delaunay triangulation algorithm to the scattered data and produce the initial triangular mesh (b), in which some triangle edges severely misalign at the boundary between the wing and the body as seen from the zoomed display in (d). After surface fitting and edge alignment, the edges of the spline triangles align with discontinuity curves in the data and can be detected as shown in (e). Finally, fine-tuning and fairing is performed to give the result in (c). Note that the whole surface is a single quadratic spline mesh. By contrast, if TP-splines were used, we would have to trim a rectangular bi-quadratic (quartic) surface into the airplane shape, or stitch-up several surface pieces; both methods are tedious.

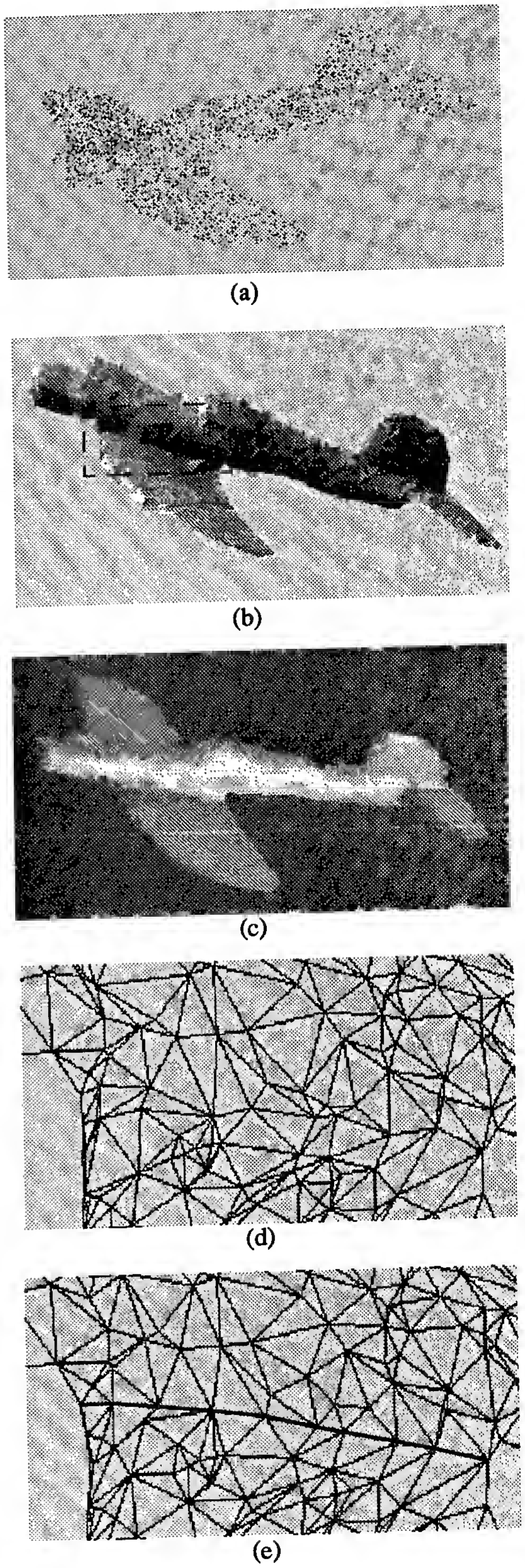
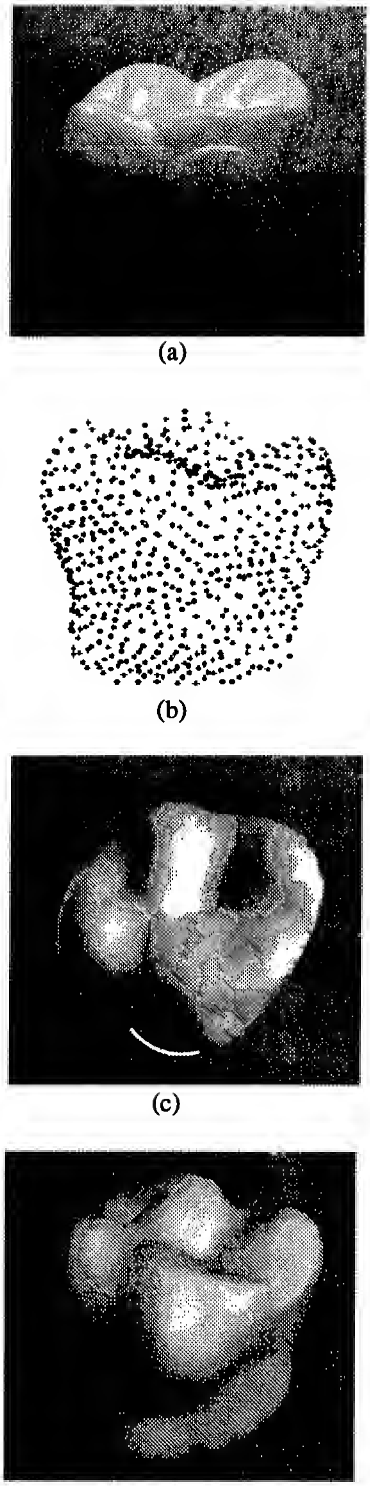


Figure 4. An airplane (open surface)



(a)

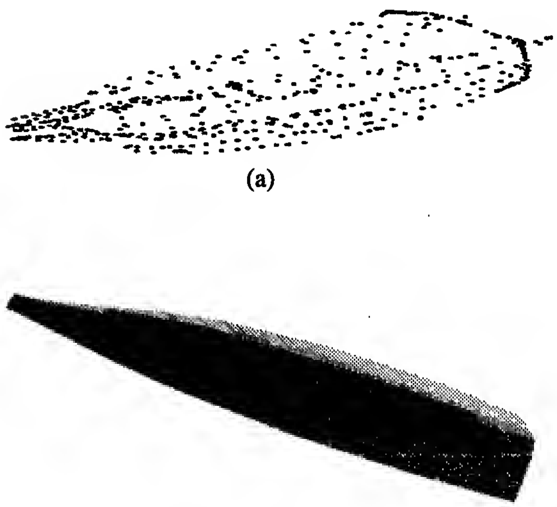
(b)

(c)

(d)

Figure 5. A tooth (closed shape)

Figure 5(a) shows the intensity image of a plaster tooth and (b) shows 650 points measured on it. We regularly subdivide a sphere into 720 triangles as the initial surface. A top view of the final C^1 surface with detected and preserved edges is given in (c). Another view is given in (d). In dental CAD/CAM, preserving the sharp edges on a tooth is very important, otherwise an upper tooth cannot align tightly with the lower tooth; on the other hand, the sides of the tooth must be smooth, or else the machined crown cannot be put on a specific patient's tooth. Our winged B-snakes represented with triangular B-splines seem very promising for smooth surfaces with embedded discontinuities. We are working on merging the triangles in smooth areas for model simplification; also we are starting to work on more complicated objects, such as mechanical parts and medical CT/ MRI data of human brains and organs.



(a)

(b)

Figure 6. A banana

The final example is a banana. Figure 6 (a) is the sampled data (about 300 points). A planar triangle region is used as an initial surface, and the final surface with detected and preserved crease edge is shown in (b).

5 Discussion

In this paper we only describe open surface and spherical surface modeling from scattered data. Compared with tensor-product splines, triangualr splines offer more flexibility. Without triangular B-splines, the surface reconstruction and embedded edge representation would become very complicated. We are in the process of extending the scheme to arbitrary topology surfaces: first trace out the topology information (a dense triangular mesh) from the potential volume fields using the Marching-cube algorithm [25], then perform mesh reduction and use the simplified mesh as an initial surface. After surface-fitting/edge-alignment with the winged B-snake paradigm, construct a G^1 overlap/blend surface with preserved discontinuities [26][27]. Local subdivision/merging, and multiresolution representations will be used for compact representation and efficient reconstruction.

6 Conclusion

In this paper we have proposed a new scheme for reconstructing surfaces from sparse noisy scattered data that may contain unspecified discontinuity edges and junctions. At first, we use a vector voting technique to infer dense surface/edge/junction potential information. Then we drop a quadratic triangular B-spline deformable surface coupled with active edges in the inferred potential fields. After some energy minimization iterations by adjusting the control points, the discontinuity edges and junctions are automatically aligned and detected, and then preserved by setting the knot configurations in constructing the final C^1 smooth surface. For tasks with high precision requirements, fine-tuning and fairing the surface by further adjusting the weights in the triangular NURBS is also effective.

References

[1] G. Nielson, CAGD Top 10: What to Watch, IEEE CG&A, Vol. 13, No. 1, Jan. 1993, pp. 35-37

[2] G. Nielson, Scattered Data Modeling, IEEE CG&A, Vol. 13, N. 1, Jan. 1993, pp. 60-70

[3] G. Guy and G. Medioni, Inference of Surfaces, Edges and Junctions from Sparse 3D Points, IEEE Int'l Symposium on Computer Vision, Florida, Nov. 1995

[4] W. Dahmen, C. Micchelli, and H. Seidel, Blossoming Begets B-Spline Bases Built Better by B-Patches, Mathematics of Computation, 1(1), 97-115, Jul. 1992

[5] P. Fong and H. Seidel, An Implementation of Triangular B-Spline Surfaces over Arbitrary Triangulations, CAGD, 10, 267-275, 1993

[6] R. Pfeifle and H. Seidel, Fitting Triangular B-Splines to Functional Scattered Data, Graphics Interface, 26-33, 1995

[7] R. Pfeifle and H. Seidel, Spherical Triangular B-Splines with Applications to Data Fitting, Eurographics, 89-96, 1995

[8] S. Auverbach et al., Approximation and Geometric Modeling with Simplex B-Splines Associated with Irregular Triangles, CAGD, 8 (1991), 67-87

[9] G. Farin, From Conics to NURBS: A Tutorial and Survey, IEEE CG&A, 12, 1992, 78-86

[10] L. Piegl, On NURBS: A Survey, IEEE CG&A, 11, 1991, 55-71

[11] L. Piegl and W. Tiller, The NURBS Book, Springer-Verlag, 1995

[12] H. Qin and D. Terzopoulos, Triangular NURBS and their Dynamic Generalizations, CAGD, 1996.

[13] Y. Shirai, 3-Dimensional Computer Vision, Chapters 3, 7 (Edge Detection), Springer-Verlag, 1987

[14] Y. Leclerc and S. Zucker, The local structure of image discontinuities in one dimension, IEEE Trans. on PAMI 9, 1987, 341-355

[15] A. Blake and A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, 1987

[16] D. Terzopoulos, Regularization of inversc visual problems involving discontinuities, T-PAMI 8, 1986, 413-424

[17] P. Alfeld, M. Neamtu and L. Schumaker, Bernstein-Bezier Polynomials on Spheres and Sphere-like Surfaces, CAGD, to appear.

[18] P. Alfeld, M. Neamtu and L. Schumaker, Fitting Scattered Data on Sphere-like Surfaces using Spherical Splines, CAGD, to appear.

[19] W. Hohenberger and T. Reuding, Smoothing Rational B-splinc Curves Using the Weights in an Optimization Procedure, CAGD, Dec. 1995, 837-848

[20] G. Dobson, W. Waggenspack, and H. Lamousin, Feature Based Models for Anatomical Data Fitting, CAD, vol. 27, no. 2, pp 139-146, 1995

[21] H. Qin and D. Terzopoulos, Dynamic NURBS Swung Surfaces for Physics-based Shape Design, CAD, vol. 27, no. 2, pp 111-127, 1995

[22] P. Laurentgengoux, M. Mekhilef, Optimization of a NURBS Representation, CAD, vol. 25, NOV, 1993, 699-710

[23] V. Theodoracatos and D. Calkins, A 3-D Vision System Model for Automatic Object Surface Sensing, (using NURBS), Int'l J. of Comp. Vision, vol. 11, no. 1, Aug. 1993, 75-99

[24] L. Creswell et al., Mathematical Modeling of the Heart using MRI, (using NURBS), IEEE Trans. Medical Imaging, vol. 11, no. 4, Dec. 1992, 581-589

[25] W. Lorensen and H. Cline, Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm, Siggraph, Jul. 1987

[26] C. Grimm and J. Hughes, Modeling Surfaces of Arbitrary Topology using Manifolds, SIGGRAPH, 359-368, 1995

[27] S. Han, Surface Reconstruction from Sparse Scattered Data, PhD research proposal, CS Dept., USC

Dynamic Programming Delineation

Lee Iverson[*]
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
E-MAIL: leei@ai.sri.com
HOMEPAGE: http://www.ai.sri.com/~leei/

Abstract

We describe a simple extensible framework for interactively delineating image curves based on user input. Like snake methods, it is based on the optimization of a potential function defined over the image, and is thus adaptable to a wide range of linear features. Unlike snakes, however, it is based on the computation of a globally optimal path between designated points. Thus results are stable, repeatable and independent of the dynamical properties necessary to ensure proper snake convergence. We will also demonstrate a variety of ways in which the dynamic programming delineation (DPD) may be extended to deal with other than purely linear features.

1 Introduction

Supporting user-guided, semi-automatic extraction of features and editing automatically extracted features from images has become a primary focus of IU research in recent years. One of the most successful of such methodologies is "snakes," in which a smooth, approximate curve is optimized to align with a designated linear image feature [Kass *et al.*, 1987]. However, even though snakes have gained very wide acceptance, they have significant problems which tend to boil down to the observation that final results are often not robustly related to the input parameters (e.g. initial curve estimates or snake dynamics). This is a common criticism of gradient descent methods, which are vulnerable to being trapped by local minima. Snakes are especially vulnerable because of their high-dimensionality.

An alternate, but related methodology has recently been rediscovered which replaces the gradient descent optimization with a dynamic programming method. Simply put, the image is considered as a planar graph with natural adjacencies and the minimal cost path is computed from one pixel to another using a dynamic programming algorithm. First described in this form as part of an automatic system for extracting road networks from overhead imagery [Fischler *et al.*, 1981], it was initially solved with a multi-pass algorithm named FSTAR. The principle was later rediscovered and recast in an interactive form in [Mortensen and Barrett, 1995], in which a variant of Dijkstra's minimal cost path algorithm was used. Largely because of the thousandfold increase in available desktop computational power, what was an off-line technology in 1981 had become a fast, interactive technique in 1995.

Subsequent research demonstrated an isomorphism with numerical methods for solving certain initial value PDE's [Cohen and Kimmel,

[*] This work was sponsored by the Advanced Research Projects Agency under contract DACA76-95-C-0010 monitored by the U.S. Army Topographic Engineering Center, Fort Belvoir, VA. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, or SRI International.

1996] and this revealed useful means of constraining the extracted curves (e.g. by bounding curvature). In addition, a search for more explicit means of constraining curvature demonstrated a way of extending the graphs beyond a simple image isomorphism to explicitly represent other properties [Merlet and Zerubia, 1996].

We present a formalization of these related methods which reveals the richness of this paradigm. By considering the similarities and differences between the alternatives already explored we propose a number of innovative uses of this technology, and avenues for further exploration.

2 General Methodology

Dynamic programming delineation (DPD) on images begins by defining a isomorphism between an image and a graph in which each pixel is associated with a vertex and the transitions between neighbouring pixels are associated with graph edges. For a discrete image $I : N^2 \rightarrow \Re$, we define a graph $G_I = (V, E)$ such that there is a one-to-one mapping from image pixels to graph vertices

$$V = \{v_i \mid i \in I\}.$$

The graph's edges correspond to pixel adjacencies in the image

$$E = \{e_{ij} \mid j \in \texttt{Neigh}(i)\}.$$

Given this graph, we formulate the delineation problem as the solution to a minimal path problem given two points $i_0, i_1 \in I$ and a non-negative potential function $P : E \rightarrow \Re^+$. The straightforward solution to this problem is a dynamic programming algorithm known as Dijkstra's minimal-path algorithm [Cormen *et al.*, 1990, p. 527]. It manages four data structures: S the set of vertices for which we know the minimal cost path from v_0 to v; an array $C[v] \in \Re$, representing this cost; $Q = \{(v, Q_v)\}$, a priority queue of vertices ordered by the partial path cost Q_v; and the array $\pi[v] \in V$ such that $\pi[u] = v$ means that u *precedes* v on the minimal path to v. The algorithm is then as outlined in Fig.1.

```
**Dijkstra** $(G, v_0, v_1)$:
  $S \leftarrow \emptyset$
  $Q \leftarrow \{(v_0, 0)\}$
  *foreach* $v \in V$ :
    $C[v] \leftarrow \infty$
    $\pi[v] \leftarrow$ *nil*
  *loop*
    *let* $(u, Q_u) \leftarrow$EXTRACT-MIN $(Q)$
      $S \leftarrow S \cup \{u\}$
      $C[u] \leftarrow Q_u$
      *foreach* $v \in \{v \mid e_{uv} \in E\}$ :
        *let* $d_v \leftarrow C[u] + P(e_{uv})$
          *if* $(v, Q_v) \notin Q$ or $d_v < Q_v$ *then*
            $Q \leftarrow Q \cup (v, d_v)$
            $\pi[v] \leftarrow u$
    *while* $u \neq v_1$.
```

Figure 1: Dijkstra's minimal path algorithm.

If this algorithm is then adjusted so that the cost used to order the queue is adjust by a lower-bound estimate of the cost from v to the goal position v_1 (typically a constant multiple of the distance $||v_1 - v||$) then the process is biased toward the goal and the algorithm is called A^* [Duda and Hart, 1973]. Using a traditional binary heap, this algorithm is $O(E \log V)$.

Since DPD is an interactive methodology, we frame this algorithm in a structure which reflects point selection and motions. In the basic form, the user initiates the process with a mouse click on the image that selects v_0. As the pointer moves, the dynamic programming algorithm is run from v_0 to v_1, the current mouse position. The display is then updated by drawing a curve over the image from v_1 to v_0 by backtracking through π. Rather than reinitializing the process throughout, S, Q, C, and π are retained as long as v_0 is constant (the costs in Q need to be readjusted if A^* is used.) Thus if $v_1 \in S$, we need not run Dijkstra at all, instead simply drawing the curve. A subsequent mouse click then freezes the location of v_1 and saves the path.

To this point, we have assumed only that the cost $P(e_{uv})$ is an arbitrary non-negative traversal cost for the edge e_{uv}. In order to actually select particular image features, we need to specialize this cost. In its simplest form, we can simply reduce this to a cost function for the terminal pixel of the edge, which we'll designate $\tilde{P}(v)$. In this special case, if we use four-connected neighbourhoods for $\texttt{Neigh}(v)$ we are left with a method similar to the Fast Marching Method (introduced by Sethian in [Sethian, 1996]) for solving for C in the initial value problem

$$||\nabla C|| = \tilde{P}, \text{ where } C[v_0] = 0.$$

The minimal path we construct by backtracking along π is then a discrete approximation to the curve $\mathcal{C}$ for which

$$\frac{\partial \mathcal{C}}{\partial s} = -\nabla C.$$

Cohen and Kimmel [Cohen and Kimmel, 1996] used this formulation to prove that if $\tilde{P}$ is of the form $\tilde{P}(v) = P(v)+w$, for $P(v)$ positive and w a positive constant, then the curve generated has the property that

$$|\kappa| \leq \frac{\sup ||\nabla P||}{w},$$

where κ is the curvature of $\mathcal{C}$. Thus w is a regularizing parameter for the curves extracted.

3 Applications

Given the restricted form of image potentials defined above, we have the ability to select for a wide range of image features. For example:

$$P(v) = |I(v) - I_0| + w$$

will select for curves that follow paths of intensity I_0. Thus setting $I_0 = \sup\{I\}$ or $I_0 = \inf\{I\}$, DPD will extract bright or dark lines respectively. This is remarkably effective for road extraction on low-resolution imagery (see Fig. 2).

If we are interested in edges, a simple image transform will be effective, with

$$P(v) = |\nabla^2 I(v)| + w,$$

tracking the zeroes of the Laplacian, the basic building block of the classic Marr-Hildreth edge operator [Marr and Hildreth, 1980]. However, this formulation has the disadvantage of being insensitive to edge orientation. We would usually prefer to select curves which have a well-defined edge orientation. The key to achieving this goal is the realization that P is really a function of the edge e_{uv}. For oriented edges then we can simply consider the cross-section of I perpendicular to $\overline{uv}$, which we'll call I_{uv}^T. A simple function which selects rising edges is then

$$P(uv) = d_{\max} - dI_{uv}^T + w,$$

where $d_{\max} = \sup\{dI_{uv}^T\}$. Following the reasoning outlined in [Iverson and Zucker, 1995], this can be made even more selective for edges by combining with other derivatives, for example

$$P(uv) = d_{\max} - dI_{uv}^T + d^3 I_{uv}^T + w,$$

where

$$d_{\max} = \sup\{dI_{uv}^T\} - \inf\{d^3 I_{uv}^T\}.$$

DPD can also, perhaps less obviously, be used as a higher-level linking method for some other low-level feature detector. This approach was adopted in a project in which the goal was to detect and delineate rivers and streams from overhead imagery [Fua, 1996]. A multi-baseline stereo algorithm was used to reconstruct detailed elevation maps over a selected image. An automatic delineation method [Fischler and Wolf, 1983] was then used to select those points M in the image at which linear structure was evident. This bitmap was then used as a mask for terrain curvature data κ to construct a cost image

$$P(v) = \begin{cases} \kappa_m - \kappa(v) & \text{if } \exists u \in M : ||u - v|| \leq \epsilon; \\ K & \text{otherwise.} \end{cases}$$

where $\kappa_m = \sup\{\kappa(v)\}$, $K \gg \kappa_m$ and ϵ is a small mask dilation (e.g. 3 pixels). As can be seen in Fig. 3, the DPD process has the effect of linking together selected points in an intuitive fashion while simultaneously choosing maximal curvature points within the masked regions. Thus we have a method for effectively and intuitively combining a logical feature selection method with a scaled measure.

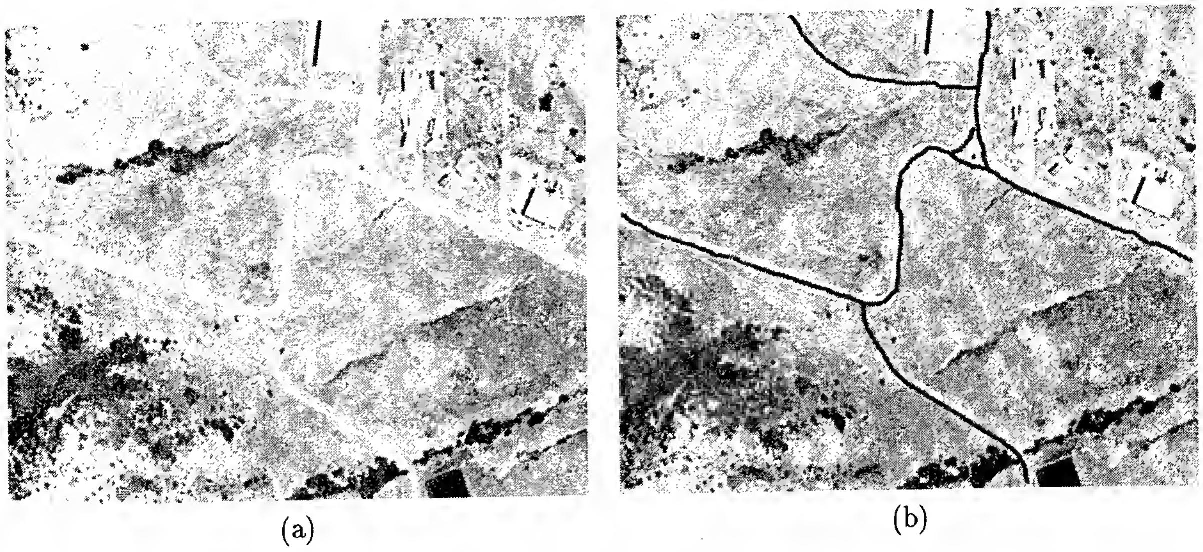


Figure 2: Low resolution road extraction using simple DPD with $I_0 = 1$. Delineation of the entire road network took approximately 10 seconds and 10 mouse clicks.

4 Extended Graphs

So far we have considered only situations in which there was a perfect isomorphism between the input image and the graph on which we are seeking optimal paths. One way in which we can loosen the constraints on the nature of the image features we select for (and thus enrich our representational ability) is to allow for extension of the graph.

This idea was first introduced in [Merlet and Zerubia, 1996], in which an explicit curvature constraint was imposed by associating eight graph vertices with each image pixel, one for each direction used to enter the pixel. Thus, using the same format as above, $G'_I = (V', E')$ such that there is a mapping from image pixels to graph vertices

$$V' = \{v_{i\lambda} \mid i \in I \wedge \lambda \in L\},$$

where $L = \{1, \ldots, n\}$, a set of labels, for some integer n. The graph's edges correspond to local neighbourhoods in the vertex set

$$E' = \{e_{uv} \mid u \in V' \wedge v \in \mathtt{Neigh}'(u)\}.$$

In [Merlet and Zerubia, 1996], the neighbourhoods were restricted so that the edges from pixel i to $j \in \mathtt{Neigh}(i)$ connect to vertex $v_{j\lambda}$ where λ was the *direction* of the path $\overline{ij}$. Furthermore, e_{uv} only existed if the direction implicitly specified by vertex u is less than 90 degrees different than the direction of e_{uv}. In this way, they were able to guarantee that paths with local bends greater than 45 degrees were impossible. As we noted above, the same condition can be imposed with an appropriate constant w added to the potential P, but the principle exposed by Merlet and Zerubia is of more general usefulness.

For example, if we wish to extend the road extraction problem suggested earlier to higher resolution imagery, then we will have to deal with the problem of road width. One way to explicitly capture the width of the road at every point along its length is to introduce a label set $L = \{1, \ldots, n\}$ where the label $\lambda \in L$ specifies a road of width $2^{\lambda/2}$. Since roads have reasonably continuous width, we will restrict the graph edges so that the edge e_{uv} only exists for $u = (i, \lambda_i)$ and $v = (j, \lambda_j)$ when $j \in \mathtt{Neigh}(i)$ and $|\lambda_i - \lambda_j| \leq 1$. Thus, even though the extended graph G'_I now has n times as many vertices as G_I, there are only three times as many edges in E', and we should still be able to run the DPD process at interactive rates. Moreover, if the potentials $P(uv)$ are calculated with respect to a pre-computed multi-resolution image pyramid, then the evaluation of the potential can be restricted to a small local neighbourhood

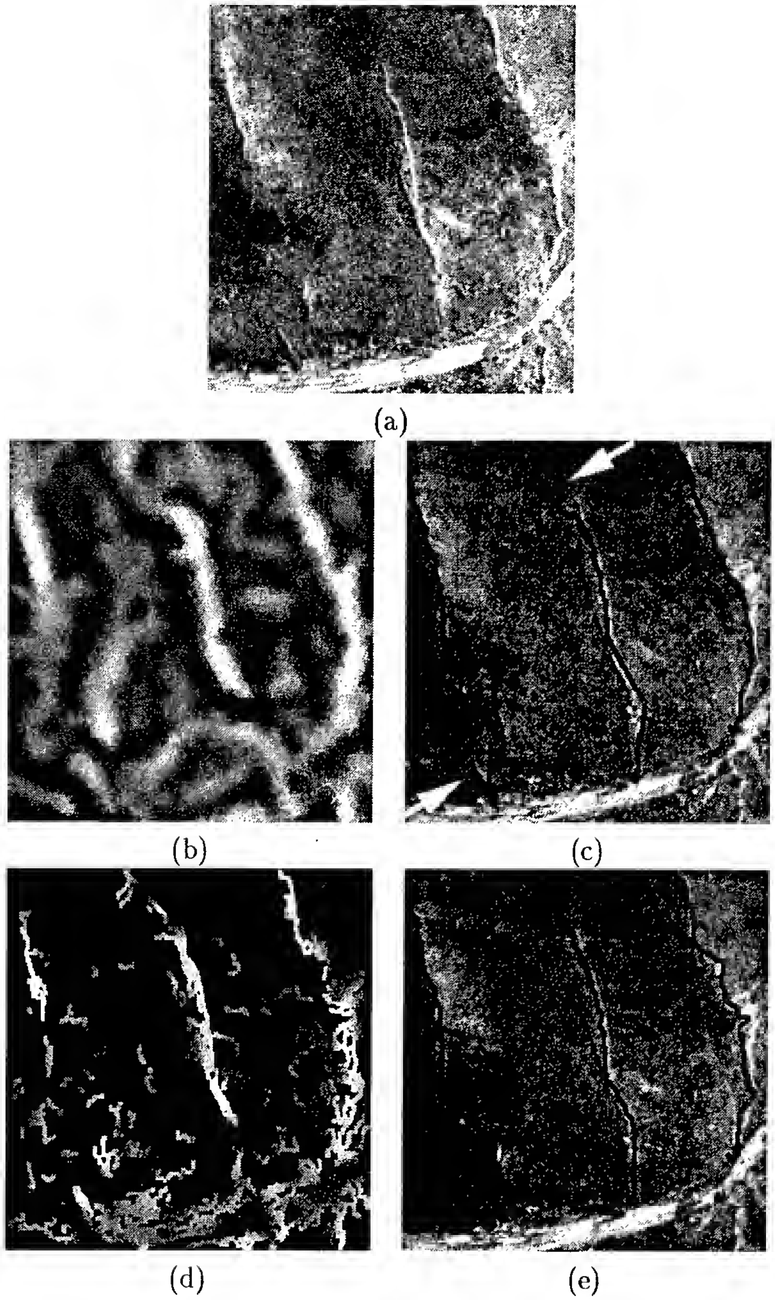


Figure 3: River delineation for three creeks flowing down a slope (a). Note that using pure curvature (b) for the delineation potential P fails to locate certain creekbeds (c). If instead the curvature is masked with a linear feature mask (d), then the delineation is accurate (e). Each creekbed was generated with two mouse clicks.

in the pyramid by simply selecting the appropriate pyramid level given λ. Thus, the system will find the minimal cost path through a scale space representation of roads.

5 Conclusion

We have described a notable body of new research which has enabled a significant improvement in the ability of analysts and others working with images to define and interactively extract linear image structures. Because the methodology uses a fast, global optimization method, it is significantly more stable than popular alternatives like snakes. Even when results are inadequate, the fast interactive framework allows for the kind of immediate interaction which is necessary to get around these kinds of problems.

This technology is in its infancy and we have outlined a few directions for future growth. Even in its current form, it is clearly useful and should find its place in standard image analysis and manipulation toolkits.

Acknowledgments

The author wishes to thank Yvan Leclerc for his help with preliminary versions of this paper and Pascal V. Fua for his extensive testing of the DPD system for extracting riverbeds.

References

[Cohen and Kimmel, 1996] Laurent D. Cohen and Ron Kimmel. Global minimum for active contour models: A minimal path approach. In *IEEE International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 1996.

[Cormen *et al.*, 1990] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Mass., 1990.

[Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, N.Y., 1973.

[Fischler and Wolf, 1983] M.A. Fischler and H.C. Wolf. Linear delineation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 351–356, June 1983.

[Fischler *et al.*, 1981] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223, 1981.

[Fua, 1996] Pascal Fua. Fast, accurate and consistent modeling of drainage and surrounding terrain. *International Journal of Computer Vision*, 1996. Submitted for Publication.

[Iverson and Zucker, 1995] Lee A. Iverson and Steven W. Zucker. Logical/linear operators for image curves. *IEEE Pattern Analysis and Machine Intelligence*, 17(10):982–996, October 1995.

[Kass *et al.*, 1987] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Compute Vision*, pages 259–268, London, England, June 1987.

[Marr and Hildreth, 1980] David Marr and Ellen Hildreth. A theory of edge detection. *Proceedings of the Royal Society of London (B)*, 207(1167):187–217, February 1980.

[Merlet and Zerubia, 1996] Nicolas Merlet and Josiane Zerubia. New prospects in line detection by dynamic programming. *IEEE Pattern Analysis and Machine Intelligence*, 18(4):426–431, April 1996.

[Mortensen and Barrett, 1995] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proceedings of SIGGRAPH*, pages 191–198. ACM SIGGRAPH, August 1995.

[Sethian, 1996] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Science*, volume 93, 1996.

Finding the Perceptually Obvious Path *

Martin A. Fischler
Artificial Intelligence Center, SRI International
333 Ravenswood Ave., Menlo Park, CA 94025 USA
E-MAIL: fischler@ai.sri.com

Abstract

This paper is primarily concerned with the problem of finding a single perceptually obvious path (POP) in an image; e.g., an isolated road in an overhead view of a desert scene, or a particular line, drawn on a piece of paper, that a person points at. We briefly describe those relevant parts of a system designed to address the general problem of automatically delineating line-like structures, but focus on the perceptual, semantic, and computational issues relevant to this particular problem.

1 Introduction

This paper is primarily concerned with the problem of finding a single perceptually obvious path (POP) in an image (or selected image window); e.g., an isolated road in an overhead view of a desert scene, or a particular line, drawn on a piece of paper, that a person points at.

In reference [5] we described an architecture (Figure 1: LD block diagram) that we have found to be both general and effective for addressing the delineation problem; it involves the following subsystems and processes:

(a) Detector/Binarizer Subsystem. Binarization of the gray-level image retaining the perceptual saliency of the linear structures (e.g., Figure 2b or 3b);

(b) Generic Linear Delineation Subsystem. Partitioning and linking the binary markers into a collection of independent (perceptually obvious) generic open paths (e.g., Figure 1c, 2c).

(c) Semantic Linear Delineation Subsystem. Splitting, semantic filtering, and relinking the generic (perceptually salient) paths to obtain semantically significant delineations. Our goal here could be to find a collection of independent paths (open, closed, or both), a linked network (with or without explicit path extraction), or to find a single "best" path.

We briefly describe relevant parts of the above system, but focus on the perceptual, semantic, and computational issues relevant to this particular problem, especially the Semantic Delineation Subsystem and our proposed solution for the final process – relinking a subset of the filtered line segments into a single POP.

2 Overall Rational and Main Problems to be Solved

Given two curves, we typically have no precise quantitative procedure for determining which of the two would be more perceptually salient to a normal human observer. By perceptually

*This work was sponsored by the Defense Advanced Research Projects Agency under contract DACA76-92-C-0008 monitored by the U.S. Army Topographic Engineering Center, Alexandria, VA. The author also acknowledges the many contributions made by Helen Wolf to the work presented in this paper. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, or SRI International.

salient we mean that, after a very brief inspection, one alternative would be chosen over the other as depicting the presence of some interesting/important natural or man-made feature, or coherent structure, in an image of a natural scene; or likely to be found first if both were judged to be coherent; or judged to be a better exemplar of some semantic category. The available ranking criterion for generic curves is largely limited to the qualitative Gestalt laws of perceptual organization [1] – proximity, closure, simplicity, similarity, good continuation (smoothness), and symmetry. In addition we might have some semantic or physical constraints that could be used to disqualify a curve from being a member of a target semantic category. For example, a curve that "doubled-back" on itself (i.e., was multi-valued for azimuth) would typically not be a valid skyline for an image taken with a horizontally held camera; or, an isolated closed curve in an aerial image would be unlikely to depict an interstate freeway. In general, at this time, we can only be expected to make gross judgments in our ranking – e.g., to find a best (perceptually salient) curve when there is really only one viable candidate in the search area.

Because of occlusions and background structure, there generally is no simple way to partition the image into curves, associated with coherent objects, that are complete and have no contamination by extraneous background content. If we tried to list or assemble all possible curves prior to ranking them, an image with as few as 20-40 curve-points would be computationally impractical to process because of the factorial growth in the possible number of curves. What is implied by the above considerations is that a single step solution to the problem of selecting a single most salient curve is probably not attainable; we must perform a sequence of grouping, filtering, and information-reduction steps to eliminate unlikely candidates as early in the selection process as possible, and then make our final selection on a greatly simplified reduction of the originally presented data.

We have examined two distinct approaches to the delineation problem in general, and to finding the POP in particular: (a) Dynamic Programming (DP) [2] which is capable of finding a least-cost path in a real-valued 2-D array (which could be the original picture, or some derived overlay called a "cost" image), and (b) a number of graph-theoretic techniques which, in practice, require an early binarization of the input image.

DP, or any other global optimization technique that can operate on the actual input data becomes computationally infeasible for anything other than cost/objective functions that are very "local" in nature. I.E., the cost of a path going through a particular pixel in an image should only be a function of an attribute list attached to that pixel and (say) the cost of appending the given pixel to a path that passes through an adjacent pixel – rather than being dependent on (say) the specific positioning of the previous five pixels in the curve segment to which attachment is being considered. Thus, the nominal generality of full global optimization is not really attainable because of computational considerations. Even if we could contend with the computational difficulties, there is the further problem of actually specifying the global cost/objective function that approximately models human perceptual behavior in interpreting graylevel images – this is an even more difficult unsolved problem.

In the approach we will now discuss, we have found (through a combination of theory and experiment – but this is primarily an empirical result) that it is possible to automatically construct a binary overlay, of almost any non-contrived graylevel image, that will retain the perceptual saliency of the linear structures (paths). It is further the case that it is now (in a binary image) possible to define the primary cues that underlie our perception of a line or path: relative proximity and smoothness of the binary (1 or 0) pixels defining the line/path. Although not a traditional Gestalt property, persistence (e.g., coherent path length) is also cue of major importance; the other Gestalt cues play a (sometimes dominant) role only when there is ambiguity due to contending interpretations, or when we recognize some known shape or repeated structure.

Generic (perceptual rather than application dependent) clustering and linking are effectively (but not perfectly) achieved by employing a modified Minimum Spanning Tree (MST) algorithm with a bound on inter-point distance. The MST algorithm we devised for this purpose can be made to run in time proportional to the number of points being processed (because the points are represented by bounded integer coordinates, their density is not arbitrary).

The result of the above steps is a collection of disjoint MST's which can be separately parsed to to provide a collection of line-segments (RPATHS) as the final output of the generic linking component of our system. This parsing process involves (1) finding a primary path through the tree (typically a diameter path), (2) trimming-back branches with ragged ends, (3) pruning short branches, (4) partitioning the remaining collection of branches into disjoint paths which are pair-wise linked at the MST nodes according to geometric and (original-image) intensity smoothness criterion. An example showing the result of this process is presented in Figures 2c and 3c.

2.1 On the Combinatorics of Finding A Perceptually Obvious Path

Assume that we start with a binarized image depicting a single POP. If we had a criterion function (CF) that allowed us to rank alternative POP candidates, we observe that the naive solution of generating and ranking all possible paths is computationally infeasible for any realistic problem. Since there are n! possible paths on n points, and $20! > 10^{18}$, a problem with as few as 20 points would be impossible to solve this way.

In general, we must address two sub-problems: (1) selecting/partitioning the actual path-points from the set of potential path-points, and (2) sequencing the selected path-points. Let us assume that we are given the points that actually constitute the solution (POP). A very reasonable CF, based on the primary Gestalt property of proximity, is density (number-of-path-points/path-length); i.e., we want to find the shortest path that contains all the given/selected points. What we have just established is that a simplification (sub-problem) of our original problem is the Traveling Salesman Problem (TSP) if the POP is closed, or the problem of finding a "Messenger" (open) path. Both the TSP and the "Messenger" path problem are known to be computationally intractable for large values of n (NP-hard). For example, (at least) until recently, the largest value of n for which there is a known solution to a non-contrived TSP was 318 cities [7][6]. While there are fast methods for finding an approximation to the solution of a Eucledian TSP problem, the perceptual character of such a solution is uncertain.

It is clear that in order to solve the POP problem we must strictly limit the the number of points that can be arbitrarily sequenced, or we must limit the number of choices that are the possible successors of any given point, or use some combination of the two preceding constraints. In a variety of problems domains that we have been concerned with (e.g., finding roads in aerial images, recognizing trees and/or finding the skyline in natural ground-level scenes), we have observed that we can usually find very dense path segments that are longer than some minimal length (related to visual detection criterion), and place perceptual and/or application-domain-related constraints on linking possibilities for these dense segments. To the extent that most of the path-points are already sequenced as members of the detected segments, and it is only the segments that must be sequenced, and even here there are only a few linking alternatives for each of the segments, we can solve the POP problem even though it is formally intractable.

Our overall-approach then is [3][5]:

(1) assemble the potential path-points into dense segments by using a fast MST algorithm (although the MST does not actually assure the densest connectivity, it usually provides a very good approximation to this condition). The input to this step is a binarized image; the output is a forest of (collection of disjoint) MST's.

(2) recover the longest segments – consistent with generic perceptual connectivity criterion –

that can be extracted from the forest of trees generated in step (1). (The list containing these segments is called RPATHS).

(3) repartition and semantically filter the collection of RPATHS to eliminate perceptual and semantic linking mistakes and irrelevant paths introduced or retained by the limited flexibility of the MST algorithm/representation and the generic parsing process.

(4) Use a very general linking technique and representation schema, capable of expressing arbitrary perceptual and semantic constraints, to imply a network of paths that is very likely to include the POP.

(5) Parse the network produced in (4) to extract a relatively small collection of prominent paths that includes the POP.

(6) Rank the paths extracted in (5), using an objective function based on the primary Gestalt criterion, and return the highest ranked path as the POP.

In the next section, we discuss some of the details of how the Semantic Delineation Subsystem (Figure 1) accomplishes steps 3 through 6.

3 The Semantic Delineation Subsystem

The Semantic Delineation Subsystem is composed of two major components; the Semantic Filter and the Semantic Linker. The Semantic Linker, in turn, has three main functional elements: (a) the SL-Segment-Linker, (b) the SL-Path-Generator, and (c) the POP-Generator.

3.1 The Semantic Filter (SF)

The purpose of the semantic filter is to extract, from a collection of perceptually salient paths, those sub-paths that are compatible with the constraints of some specified application or purpose (e.g., sub-paths that could be road segments in an aerial image).

This system component takes as its input a list of generic perceptually-salient paths (RPATHS) and produces, as its output, a list of path-segments (RPATHS-F). Each item (called a seg) in RPATHS-F, is a coherent sub-path of some path in RPATHS; the segs returned in RPATHS-F are open and non-self-intersecting, and any pair of segs are disjoint with the possible exception of a single intersection-point (as are the paths in RPATHS).

The SF processes each path in RPATHS independently. It first partitions the path into adjacent segs at it's salient points using the algorithm described in [4]. This partitioning step is necessary to recover components of the application relevant paths that were combined with other (incidental) adjacent paths in the original image. Each seg is evaluated for compatibility with the constraints of the intended application on an accept or reject basis. The accepted segs are appended to the output-list RPATHS-F. In addition, if two accepted segs were part of the same input (RPATHS) path, but are now separated in the sense that some portion of the input path between them was deleted by the filter, then an entry recording this fact is made on a link-list (see discussion of the Semantic Linker).

While the SF might have to be completely redesigned for each new application, we have found that the same set attributes (properly parameterized for the different applications) appears adequate for such diverse tasks as finding roads or rivers in aerial images, and for finding man-made objects (e.g., building edges) or natural objects (e.g., the skyline, tree-trunks) in ground-level images.

The attributes we currently evaluate (to be described in detail in a later version of this paper) are concerned with length, directionality, smoothness, and degree of randomness:

(1) Length. Very short segs are typically rejected as being "noise" or unimportant (they can be recovered later if necessary); very long segments are typically accepted since they are too important to discard without the further analysis to be performed later.

(2) Consistency of global direction based on a histogram of the directions between adjacent seg pixels obtained from a chain-coded representation of the seg.

(3) Smoothness. This property is measured in two ways. First, each seg is inherently smooth to some degree because its parent in RPATHS was partitioned into segments at salient (or high curvature) points. Thus, the length of the seg is an indirect measure of its smoothness (the longer the seg, the smoother it is). Second, we measure the seg's deviation from a best fitting circular-arc to look for a smoothness property that is especially important for some applications (e.g., finding man-made objects).

(4) Randomness. We have devised a weak measure of symmetry, or of repeated structure, in a path; this measure together with the evaluation of coherent length, consistent direction, and smoothness, provide a basis for judging whether a seg is a "purposeful" or an apparently random structure.

An example of the performance of a semantic filter we designed for delineating roads in aerial images is shown in figures 2d and 3d. Tables 1 and 2, in the section on experimental evaluation, presents quantitative results of the filtering operation in terms of a relevant set of semantic categories.

3.2 The Semantic Linker (SL)

The purpose of the semantic linker is to combine all the segs in the list RPATHS-F (produced by the Semantic Filter) into either a network of partitioned or unpartitioned paths, or to select and sequence a subset of the segs in RPATHS-F into a single POP; the problem of producing an unpartitioned network (generally, the more useful of the available types of output since a distinguished POP might not even exist) is a very simple sub-problem of producing a POP.

The SL has three components, (a) the SL-Segment-Linker, (b) the SL-Path-Generator, and (c) the POP-Generator.

3.2.1 The SL-Segment-Linker (SLSL)

The input to SLSL is RPATHS-F, and its output is the "link-pair-list." The SLSL examines every pair of segs in RPATHS-F and determines if they can be adjacent components of an extended path compatible with the constraints of the specified application. If so, it generates a "link-pair" entry which is appended to the "link-pair-list."

The SLSL typically uses three types of criteria to make a link decision for a pair of segs:

(1) The relative geometric positioning and separation of the segs. For example, in the case of road delineation, the criterion is typically a bound on the separation-distance between nominally corresponding endpoints (one on each seg). In the case of skyline delineation, the segs might be further constrained not to have any overlap in their horizontal (x) coordinates.

(2) Global attributes of the segs. For example, in the case of road delineation we might require that the spectral distribution, or image intensity, or mean width of the two candidate segs be identical to within some specified tolerance.

(3) Acceptance by the semantic filter. If the two candidate segs are linked as proposed and treated as a single seg, a sufficient condition for linking is that the combination is accepted by the semantic filter.

3.2.2 The SL-Path-Generator (SLPG)

The input to the SLPG is the "link-pair-list" produced by the SLSL and augmented by additional link-pairs supplied by the Semantic Filter; the output is either an unsegmented network (actually, a disjoint collection of such networks) or a pair of lists containing all possible maximal open-paths and loops implied by the link-pairs. The POP is assumed to be one of these (explicit) paths, and a simple test is proposed as a way of selecting it.

The function of the SLPG is purely syntactic/algorithmic – to expand the path information implicit in the augmented link-pair-list. The link-pairs are a compact encoding, actually generators, of the network or collection of paths to be produced by the SLPG.

We can easily partition a collection of link-pairs into disjoint subsets so that every pair of link-pairs referring to a common seg are in the same subset called a "link-pairs-association-set." The collection segs corresponding to such a subset is called a "seg-association-set." Each seg-association-set implies a disjoint network of paths (e.g., roads); networks consisting of a few short isolated paths can often be discarded as noise. The larger networks are typically returned as one of the major end-products of the system described here when used to find all the salient paths (e.g., roads or rivers) in an image. In this paper we are primarily concerned with a second type of output: explicitly extracting the single most salient path (the POP).

The SLPG operates as follows: The link-pairs are first partitioned into disjoint subsets (link-pairs-association-sets); these subsets are then processed independently to extract their implied paths using a collection of algorithms. For the purposes of this paper we describe the LP-Basic-Path-Extension-Algorithm (BEA) in some detail, but only indicate the basis for the remainder of the full extraction process (see Appendix).

A maximal-path through an LP-network is one that cannot be a proper continuous subsequence of some longer path; we will call the endpoints of a maximal-path terminal-nodes. A loop is a maximal-path that begins and ends on the same terminal-node. One requirement of the SLPG is to explicitly list all maximal-paths.

If the BEA is given a terminal-node as a seed, it will iteratively generate all the maximal-paths that have the given terminal-node as (at least) one of their endpoints. It is both fast and simple to find all the free endpoints (nodes of degree one) of an LP-network given its associated list of link-pairs; each such free endpoint (called an ept) is a terminal-node of one or more of the maximal-paths. In an LP-network without loops, we can generate all the maximal-paths using the set of epts as seeds. Each maximal-path will be found twice, but this redundancy does not cause any problems. The redundant paths can be avoided at considerable additional complexity in the BEA algorithm, but it is simpler to just detect and delete them should this be necessary.

If the network contains loops, except for some unusual situations, the above procedure will still return all the maximal paths (including the loops). Each loop could be generated many times (an upper bound is the product of the number of epts and "entry-points" to the given loop). If we wish to be assured that all the maximal-paths are found, and also reduce the redundant discovery of the same loop, then we can proceed as above (if there are any epts, otherwise, pick any node as the first seed and later discard the initial set of non-maximal-paths). All terminal-nodes which are not already members of the list of seeds are added to that list whenever they are found. When a loop is returned by the BEA, we have to modify all the link-pairs that point to segs that are components of the loop. We inactivate all link-pairs that point to two loop-segs, and replace each link-pair that points to exactly one loop-seg with two new link-pairs; in one case the original loop-seg-link-atom is replaced by a link-atom that will insert into a non-terminating path that originally included one or more loop-segs a dummy-seg identifying the loop; in the second case the original loop-seg-link-atom is replaced by a link-atom that identifies itself as a terminal-node associated with the given loop. In a sense, we collapse the loops in the original LP-network and create a modified loop-free network in which the BEA (algorithm) is assured to return all the maximal-paths. Those returned maximal-paths that contain dummy-segs are easily rectified.

In summary, there are some interesting theoretical issues that must be addressed in order to understand how to make the SLPG more efficient, but the algorithm we have described is computationally acceptable, and it returns all the maximal-paths as required to allow the SLPG to correctly perform its function.

3.2.3 The POP-Generator

The list of maximal-paths (both open-paths and loops) returned by SLSL is assumed to contain the POP. The segments comprising these paths have been previously filtered to assure compatibility with the semantic constraints of the specified problem domain, and are perceptually salient with respect to (at least some of) the Gestalt laws of perceptual organization. In the present algorithm, the POP-generator does not alter any of the maximal paths but simply selects the one that maximizes a combination of both path-density and path-length. Actually, the product of path-length and $(\textit{path-density})^2$ where path-density is measured by (number-of-path-points)/(path-length).

4 Experimental Evaluation

In addition to a significant amount of previous informal testing and evaluation (some parts of the Linear Delineation System were applied to well over 1,000 images of different types and with different delineation goals), we are now engaged in developing a formal evaluation methodology, especially in regard to road delineation.

In a typical road delineation problem (Figure 2) the Delineation System was invoked without any manual intervention or parameter tuning. We started with a 768X638 pixel image (489,984 points) that resulted in a binarized version (step 1) with 55,480 potential road points (Fig 2b). As a result of the generic delineation process (step 2), we extracted 340 segments (RPATHS) containing 21,255 points (Fig 2c). We defined six semantic categories of interest (Narrow Road, Wide Road, Proto Road, Ambiguous, Background, River) and manually classified the pixels along the paths into these six categories. If the labeling of a given Rpath was mixed, we counted the contiguous segments with the same label as being distinct – thus we judged that there really were 375 semantically distinct segments containing 21170 points comprising the 340 actual RPATHS with an associated count of 21,255 pixels. (Because of double counting of segment and path intersection-points, there is a small discrepancy in the number of points in the actual Rpaths and in the semantically labeled segments).

Table 1 and Fig 2d show the effectiveness of the Semantic Road Filter in retaining road points/segments while eliminating the unwanted background and river points/segments. Since the Road Filter was designed to retain narrow road segments, other structures (wide-roads, proto-roads, ambiguous) that could possibly be roads were considered to have a "don't-care" status in our evaluation.

A "window" (Figure 3) was manually selected and extracted from Figure 2 to test the POP-delineation algorithm. Here we started with a 475X149 pixel image (70775 points) that resulted in a binarized version (step 1) with 7214 potential road points (Fig 3c). The extracted set of 23 RPATHS contained 3696 points (Fig 3d). Table 2 and Fig 3e show the result of applying the Semantic Road Filter; it returned 37 segments containing 3117 points in RPATHS-F and 22 link-pairs (in *aux-link-pair-list*). The SL-Segment-Linker produced 19 additional link-pairs, and thus a total of 41 distinct link-pairs were supplied to the SL-Path-Generator; these were classified as consisting of 6 ept-pairs, 34 interior pairs, and 1 closed pair. The SLPG returned 43 open paths and 115 paths containing loops; a total of 158 maximal-paths. This set of paths contained redundant entries; actually, there were 8 distinct open-paths and 4 distinct closed-paths (loops). Each path was assigned a ranking using the the product of path-length and $(\textit{path-density})^2$ metric presented in the preceeding section. The POP-generator then selected the highest ranking path (it happend to be one of the closed-paths) as the POP (Fig 3f). This was the desired delineation.

5 Discussion

The work described in this paper is part of an on-going effort to fully automate the process of delineating perceptually and/or semantically meaningful line-like structures appearing in both aerial and ground-level images of scenes consisting mostly of natural features (e.g., trees, vegetation, drainage, and terrain) as well as some man made objects (especially roads). Our intent in preparing this paper was, in addition to its nominal subject matter, to describe relevant components of the system being assembled

	RPATHS			RPATHS-F2		
Category	# points	% points	# paths	# points	% points	# paths
Narrow Road	5843	28	45	5322	46	40
Wide Road	2102	10	12	2082	18	12
Proto Road	2941	14	62	1671	15	46
Ambiguous	1579	7	37	425	4	19
Background	8192	39	210	1720	15	87
River	513	2	9	294	3	5
Total	21170	100	375	11514	100	209

Table 1: Categorized Delineations for FT-HOOD1 image. Total pixels = 768 x 638 = 489984

	W1-RPATHS			W1-RPATHS-F2		
Category	# points	% points	# paths	# points	% points	# paths
Narrow Road	3060	85	8	2935	94	7
Wide Road	0	0	0	0	0	0
Proto Road	63	2	1	46	1	1
Ambiguous	146	4	3	46	1	3
Background	319	9	10	88	3	6
River	0	0	0	0	0	0
Total	3588	100	22	3115	100	17

Table 2: Categorized Delineations for FT-HOOD1-W1 image. Total pixels = 475 x 149 = 70775

for this purpose, illustrate and quantify some of its current performance, and discuss some aspects of the conceptual basis for its design.

The problem of finding the POP in (some designated portion of) an image is a basic requirement for effective man-machine communication about images, as well as a challenging problem whose solution is required to accomplish some of the more general delineation tasks. In this paper, we provide an approach to the solution of this problem, and an algorithm that is applicable to a limited class of scene domains. The algorithm has performed well on a small set of test cases but a significant amount of additional testing will be required before be can be sure of its utility and robustness.

An important contribution of this paper is the introduction of the LP-representation (link-pair/LP-network) and associated machinery as a generalization of the conventional graph. The LP-network provides a very powerful way of dealing with linear structures; it provides almost complete generality in specifying connectivity (more than is possible with a graph), it provides a very compact description of the (implied) connected structures, and it admits reasonable algorithms for the common (relatively simple) situations to be expected in images of real scenes. On the other hand, because there are specializations of the linking problem that are *NP-hard*, there are no generally efficient algorithms for this purpose.

There are many open problems and obvious extensions of the work discussed in this paper. However, one of the more interesting extensions would be to find a way to duplicate human performance in the following type of situation:

Consider an image composed of a sequence of 50 equal signs typed in a row (i.e., ====================== ...). Also assume there is a solid horizontal line positioned just below the equal signs that has the same horizontal extent. If we assume that there are four links possible between each pair of successive equal signs (two straight links and two cross-over links; we ignore the vertical links which lead to short closed paths), then there are on the order of 2^{50} paths that potentially would have to be generated before we could decide – by applying some objective function to an explicit descriptions of the competing paths – if some path through the equal signs, or the solid line, or neither, was the POP. Obviously, the human doesn't do this; he picks the solid line almost immediately; how is he able to avoid the combinatorial explosion??

One of our main points in this paper, and the basis of our approach, was that much of the assembly of the ultimately to be selected POP had to take place in the generic perception phase which sacrifices flexibility and generality for simplicity and speed. The Semantic Linker is computationally limited and can't be handed a problem with too many choices. Thus, the overall delineation system must include mechanisms that enforce complexity constraints on the output of each of the its subsystems – this type of control could be accomplished by iteratively adjusting algorithm parameters. It thus appears that a vision system must be fully cognizant of its computational limitations if it is to operate effectively. Understanding how to accomplish this type of control is one of our more immediate goals.

A Appendix

A.1 Definitions

EPT: a "free endpoint" designates the end of a seg which is not referenced by any currently active link-pair; e.g., a path containing an ept cannot be further extended at that end.

LINK-ATOM: a list of two items, the first is an index number into RPATHS-F; i.e., it points to a seg in RPATHS-F. The second item is a logical variable (T or NIL) which specifies whether the seg is to be used as stored (NIL) or reversed (T).

LINK-LIST: a list of two or more link-atoms. It specifies how to assemble a path from a subset of the segs stored in RPATHS-F.

LINK-PAIR: a list of two link-atoms. It specifies a path consisting of the concatenation of the two segs in the order listed, with the points in each seg taken as stored in RPATHS-F, or reversed, as specified by the logical variables.

LINK-PAIR-LIST: nominally, the list of link-pairs produced by the SL-Segment-Linker and the Semantic Filter.

LOOP: a subsequence of a path that begins and ends with an identical link atom, or, a subsequence of a path that begins and ends with a link atoms pointing to the same seg, but having reversed directions.

LP-NETWORK: the collection of paths implied by a collection of link pairs.

CONNECTED-LP-NETWORK (CLPN): a collection of link-pairs can be partitioned into disjoint subsets so that every pair of link-pairs referring to a common seg are in the same subset called a "LINK-PAIRS-ASSOCIATION-SET." The collection segs corresponding to such a subset is called a "SEG-ASSOCIATION-SET." Each seg-association-set implies a disjoint network of paths called a CLPN.

PATH: a concatenation segs (segments) as specified by a link-list. No seg can appear more than once – with the exception of the seg specified by the head link-atom in the case of a loop or semi-loop. The HEAD of the path is intended to refer to the end at which the path is being extended; the TAIL of the path is intended to refer to the end of the path containing the seed link-atom, and we arbitrarily assume that the path is, or was, constructed by a sequential accumulation of segs starting at the tail-end. For most purposes, we further restrict this definition to prohibit the path from visiting a vertex more than once.

OPEN-PATH: a path that does not contain a (complete) loop.

MAXIMAL-PATH: A maximal-path through an LP-network is one that cannot be a proper continuous subsequence of some longer path; we will call the endpoints of a maximal-path TERMINAL-NODES. A LOOP is a maximal-path that begins and ends on the same terminal-node.

A.2 Some Attributes of an LP-network

1. If an LP-network has no loops, then the terminal-nodes of every maximal-path are epts; i.e., every fully extended path will begin and end at an endpoint of a seg which is not connected (by an active link-pair) to any other seg. If the LP-network does contain loops, then the "entry-points" to the loops will also be terminal-nodes of maximal-paths.

2. There might not be any single path connecting two given epts, even in a connected-component of the network. For example, consider a network consisting of three segs connected as a Y. If the two upper arms of the Y both connect to the lower vertical stem, but not to each other, then there is no direct path linking the two free ends of the upper segs of the Y.

3. An LP-network can always be converted into a conventional graph by adding additional link-pairs so as to make every vertex "fully-connected."

A.3 Basic Algorithms

Key algorithms include:

- lp-basic-path-extension-algorithm

- partition-lp-network-into-connected-subnetworks

- identify-ept-lp-algorithm

A.3.1 LP-BASIC-PATH-EXTENSION-ALGORITHM

For a given subset of link-pairs (say set q), a link-atom (say s1) is selected from one of the link-pairs in q as a seed, and a collection of paths are iteratively constructed from this seed by successively scanning all the link-pairs (in q) for additional segs to append to the set of paths still being extended. We note that a path is represented by a link-list (a list of link-atoms); all the partial paths we are currently extending have one endpoint defined by the initial seed s1. Consider a particular path (say p7) which has as its current terminal link-atom the list (seg24 t). If some link-pair in q has the form ((seg24 t) (segX t/nil)), then p7 can be extended by appending link-atom (segX t/nil) to its current link-list. There may be more than one link-pair in q capable of extending the current version of p7. Further, if some link-pair in q has the form ((segZ nil/t) (seg24 nil)), this link-pair representing the concatenation of segs segZ and seg24, is equivalent (in that the two segs are joined in the same way) to the link-pair represented by ((seg24 t) (segZ t/nil)), and thus p7 could be extended by appending link-atom (segZ t/nil).

At each iteration we start with a list of partial paths, and for each such path there are three possibilities: (a) there are no further extensions, in which case the path is placed on the output open-path-list; (b) there is an extension, but the extending seg (or the unattached vertex of the seg) already appears in the partial path, in which case the extended path is placed on the output closed-path-list; (c) there are one or more (single seg) extensions with new segs, in this case, all such extensions are placed on a new partial-path-list and the above process is repeated. The process terminates when the partial-path-list has no entries at the start of a new iteration.

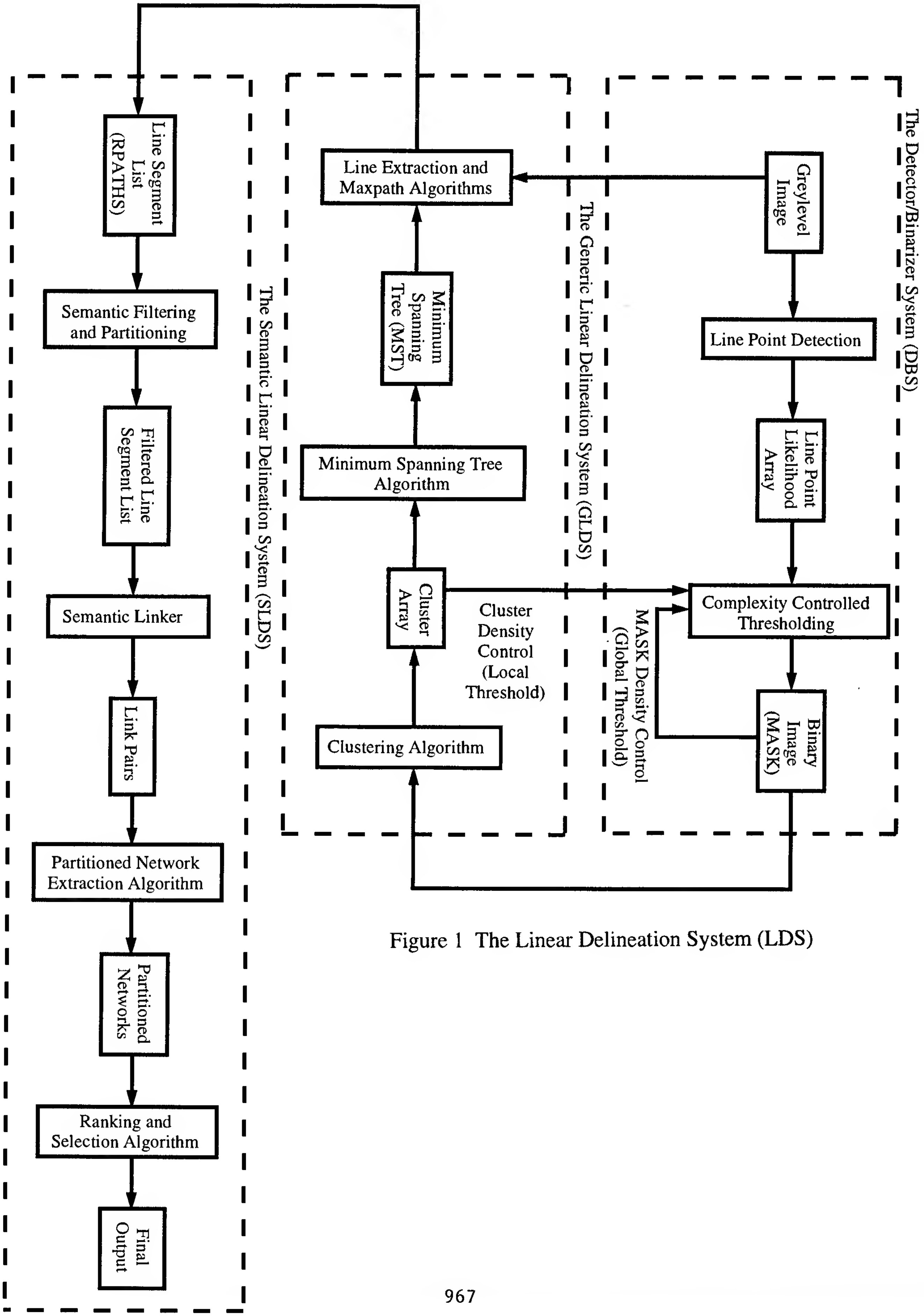
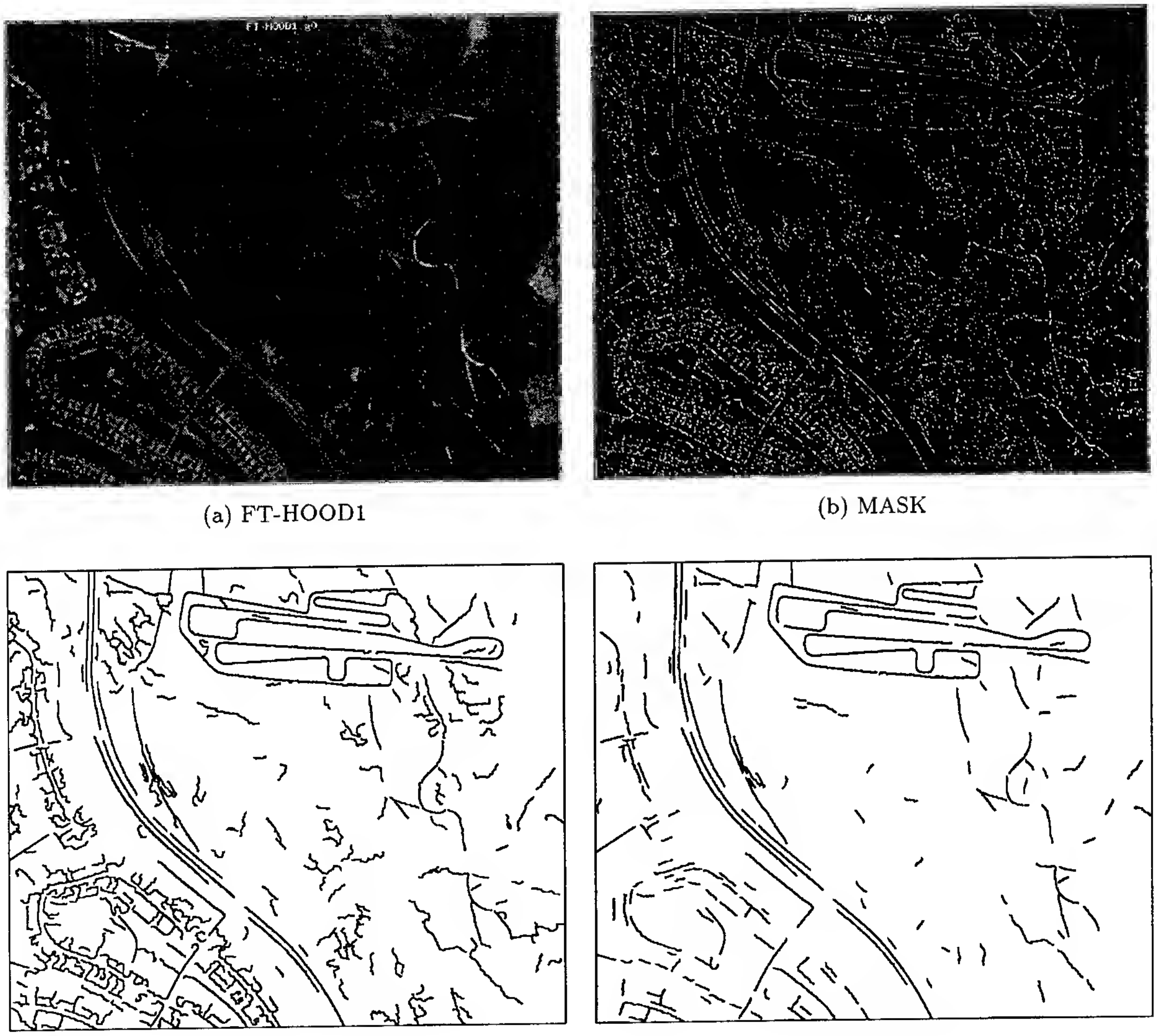

The Detector/Binarizer System (DBS)
Greylevel Image
Line Point Detection
Line Point Likelihood Array
Complexity Controlled Thresholding
Binary Image (MASK)
MASK Density Control (Global Threshold)
The Generic Linear Delineation System (GLDS)
Line Extraction and Maxpath Algorithms
Minimum Spanning Tree (MST)
Minimum Spanning Tree Algorithm
Cluster Array
Cluster Density Control (Local Threshold)
Clustering Algorithm
The Semantic Linear Delineation System (SLDS)
Line Segment List (RPATHS)
Semantic Filtering and Partitioning
Filtered Line Segment List
Semantic Linker
Link Pairs
Partitioned Network Extraction Algorithm
Partitioned Networks
Ranking and Selection Algorithm
Final Output

Figure 1 The Linear Delineation System (LDS)



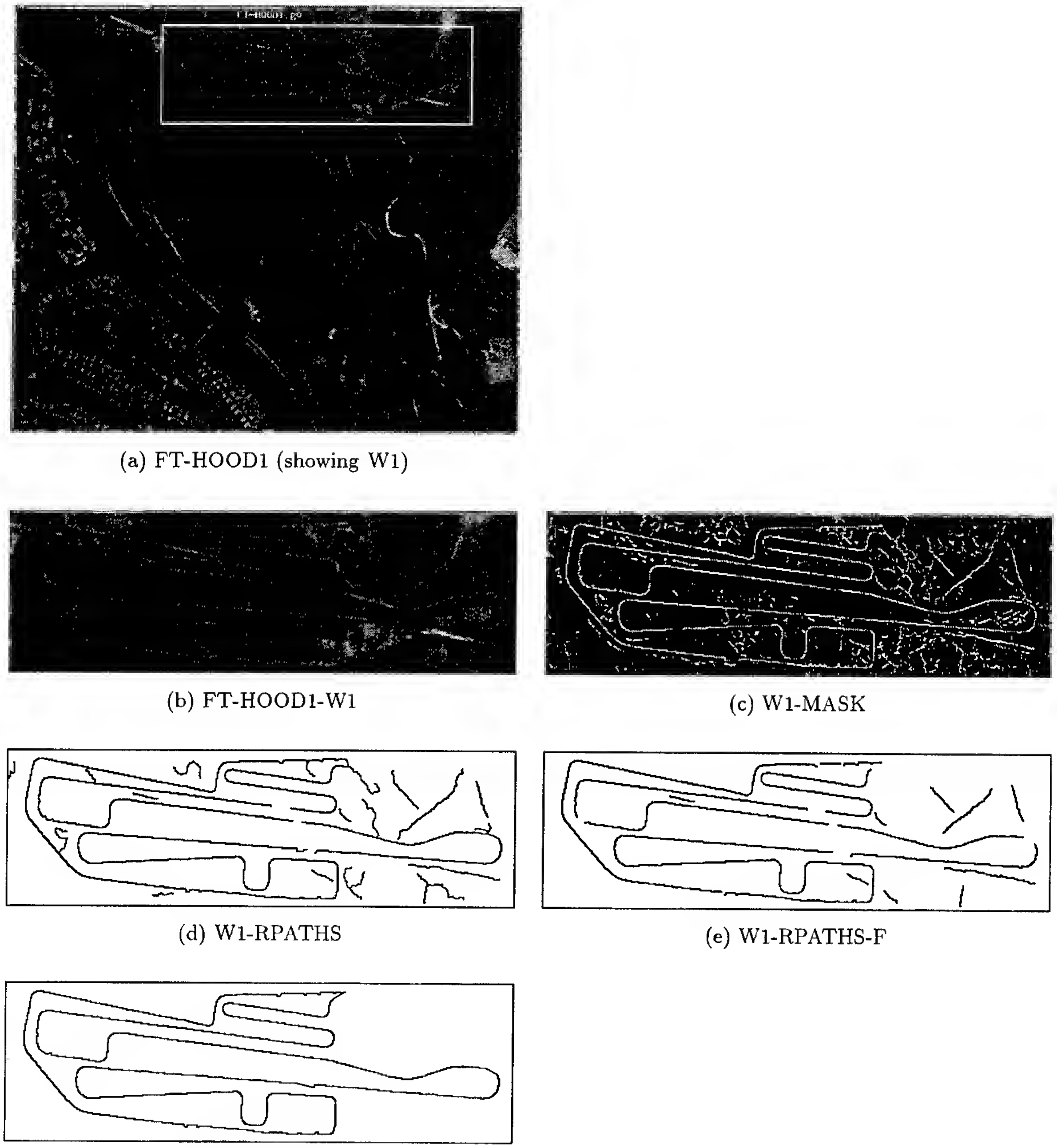
(a) FT-HOOD1

(b) MASK

(c) RPATHS

(d) RPATHS-F

Figure 2: Extracting Roads from an Aerial Image



(a) FT-HOOD1 (showing W1)

(b) FT-HOOD1-W1

(c) W1-MASK

(d) W1-RPATHS

(e) W1-RPATHS-F

(f) W1-RACETRACK

Figure 3: Extracting a POP from an Aerial Image

References

[1] M.A. Fischler and O. Firschein, "Intelligence: The Eye, the Brain, and the Computer," (331 pgs.), Addison Wesley, 1987.

[2] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique," Computer Graphics and Image Processing, vol 15(3), March 1981, pp 201-223; also, Readings in Computer Vision (M.A. Fischler and O. firschein, eds.), Morgan Kaufmann, pp 740-752, 1987.

[3] M.A. Fischler and H.C. Wolf, "Linear Delineation," Proceedings IEEE CVPR-83, June 1983, pp 351-356; also, Readings in Computer Vision (M.A. Fischler and O. firschein, eds.), Morgan Kaufmann, pp 204-209, 1987.

[4] M.A. Fischler and H.C. Wolf, "Locating perceptually salient points on planar curves," IEEE PAMI vol. 16(2):113-129, Feb. 1994.

[5] M.A. Fischler, "The Perception of Linear Structure: A Generic Linker," Proc. ARPA Image Understanding Workshop, Monterey, Calif, Nov, 1994.

[6] A. Gibbons, "Algorithmic Graph Theory," Cambridge Univ. Press, 1985.

[7] E.L. Lawler, et. al., "The Traveling Salesman Problem," Wiley-Interscience, 1985.

Knowledge Directed Reconstruction from Multiple Aerial Images*

Christopher O. Jaynes, Mauricio Marengoni, Allen Hanson,
Edward Riseman, and Howard Schultz

Computer Vision Laboratory
Dept. of Computer Science
Box 34610, University of Massachusetts
Amherst, MA. 01003
E-mail: `jaynes@cs.umass.edu`, URL: `http://vis-www.cs.umass.edu`

Abstract

Image understanding (IU) techniques for automatic site reconstruction have demonstrated success within restricted domains and for small numbers of model classes. However, these techniques often fail when applied out of context and do not "scale-up" into a more general solution. Under the APGD program, we are constructing a knowledge-based site reconstruction system that automatically selects the correct algorithm according to the current context, applies it to a focused subset of the data, and constrains the interpretation of the result through the explicit use of knowledge.

1 Introduction

The extraction and reconstruction of building models from aerial images has become an important area of research in recent years. Significant progress has been made and several systems perform reasonably well within their appropriate domains [Collins'95, Herman'94, Lin et al.'94, Chellapa et al.'94]. For example, recent testing of the Ascender I system has shown it capable of automatically extracting a large percentage of the buildings within a subregion of the Fort Hood dataset [Collins, et al'96]. Although these results are significant, the system was designed to perform well under particular contexts and is only capable of detecting the single class of buildings whose rooftops are flat rectilinear polygons.

The modest successes attained by Ascender I and similar systems can, we believe, be traced to their narrow scope and application to highly constrained data. The class of flat roofed rectilinear buildings is very clearly defined by a set of geometric and spatial properties which are useful for recognition if the incidence of distracting classes is small (that is, when the data is suitably constrained). This idea of a (set of) local expert(s) for recognizing instances of an object class played a prominent role in early work on the Schema system [Draper'89], as well as other systems in the aerial image domain [Chellapa et al.'94, Huertas and Nevatia'80, Gifford and McKeown'94, Jaynes'96a, Matsuyama'85]. Under this model, robustness is achieved by providing multiple reconstruction/recognition strategies which are applicable under well defined conditions and generality is achieved by increasing the number of object classes to describe a larger fraction of the world.

Work has begun on Ascender II, a geometric site modelling system based on this general framework. The design of Ascender II is founded on three basic principles: 1) Specific image understanding strategies are clearly successful under particular contexts for a particular class of objects but may break down when applied in contexts that exceed the design constraints. 2) Domain knowledge, knowledge acquired from partial processing of the data, and knowledge about available image understanding strategies are all valuable in constraining the reconstruction problem. 3) A successful system will contain many specific strategies but will selectively apply them in the correct context, with the correct set of parameters, and will fuse the results of individual strategies into a complete reconstruction.

2 Ascender II

The Ascender II system explicitly represents both knowledge and context to support a purposeful reconstruction of the site using geometric and spatial reasoning and intelligent control of sophisticated IU algorithms. The system is divided into a vi-

*Funded by the RADIUS project under DARPA/Army TEC contract number DACA76-92-C-0041, by NSF grant number CDA-8922572, and in part by the National Council for Scientific Research-Brazil grant number 260185/92.2.

sual subsystem and a knowledge base. The *visual subsystem* resides within the Radius Common Development Environment (RCDE) [Mundy et al.'92] and contains a library of IU algorithms, a geometric database that contains available data (images, line segments, functional classifications, etc.), as well as models that may have been acquired through processing. The *knowledge base* is based on belief networks and is constructed using HUGIN [Andersen'89], a system for designing belief networks and influence diagrams. The knowledge base consists of reasoning mechanisms, a control system, and the belief network that represents the current set of knowledge about the site. The two systems communicate through Unix socket IP mechanisms. Figure 1 shows an overview of the system.

Reasoning takes place over *regions of discourse* that represent a subset of the available data. Regions of discourse may be image regions, a particular building model, or other sets of data that may have been produced by the system.

Processing of the data proceeds in a straightforward way. First, the knowledge base is consulted and an appropriate IU algorithm and subset of the current region are selected. For example, a search for line evidence along the center of a building region may be invoked to gather evidence for the presence of a peaked roof building. The choice of algorithm and subset of the data is sent as a request to the visual subsystem for processing. The IU algorithm is applied to the data and the database is updated with the result (for example, a set of line features may be produced). The visual subsystem then converts the result into a single value that represents the belief that the requested evidence was present. This belief is passed to the knowledge base where it is used to update the belief network. The next appropriate action is then selected based on the control policy.

2.1 Knowledge Base

The knowledge base is capable of representing the current context, specific site knowledge (either engineered or acquired as part of processing), and general domain knowledge relevant to site modelling. Knowledge is stored in a *Schema Network*. The representation is a combination of two important ideas drawn from the field of Artificial Intelligence; Schemas [Draper'89] and Belief Networks [Jensen'96]. The network encodes information about how and when algorithms can be applied in the current context and explicitly represents the causal dependencies found in a particular domain.

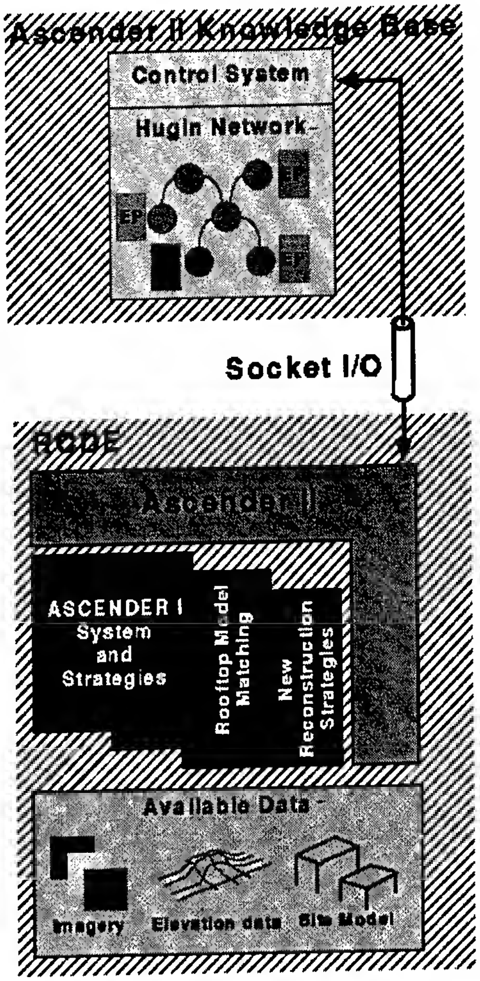


Figure 1: Ascender II system overview. Control decisions are based on the current knowledge about the site. Vision algorithms, stored in the RCDE, gather evidence about the site and produce a site model. Ascender I provides one set of IU strategies relevant to site reconstruction.

Nodes within the network represent discrete variables that are associated with the domain. For example, the node `Building-Roof-Shape` may have the discrete states {`flat`, `peaked`, `curved`, `composite`}. At each node, an *evidence policy* contains information about how evidence for a peaked roof building may be acquired. Contextual rules, part of the node's evidence policy, assist in the selection of the correct algorithm, data, and parameters, for the given context.

Edges within the network represent a conditional dependence between a parent and child node. Associated with each node is a conditional probability table that contains a probability for each state of the node given the values of the parents. The *Belief* for a parent node, then, can be computed from the values of the children using causal influence [Russel'95]. As evidence is added to the network (through the execution of an evidence policy), the effect is propagated throughout the network and a new set of belief values are computed.

2.1.1 Action Selection

The problem of action selection within the schematic network is a significant one. Currently we take a greedy approach. In order to gather evidence about a node n, the corresponding evidence policy is invoked. If the evidence policy at n is empty (there are no IU algorithms directly applicable to computing belief(n)) or there are no available algorithms for the current context, then the children of n are visited. The node whose belief value contains the highest uncertainty is selected and either its evidence policy is invoked or its children are visited. Once evidence has been computed, the belief values are propagated back through the network and a new action is selected. This implies that there must be at least one evidence policy available at each of the leaf nodes within the network.

Certainty for node n is defined as difference of the maximum belief and the belief value if all states at n are equally likely.

$$max(Belief(n) - \frac{1}{(Num\ states)_n})$$

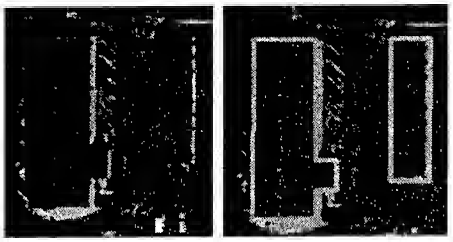
2.2 Visual Subsystem

The visual subsystem is comprised of two parts; a function library that stores the set of IU algorithms available to the system, and a geometric database that contains available data in the form of imagery, partial models, and other collateral information about the site (such as classification of functional areas).

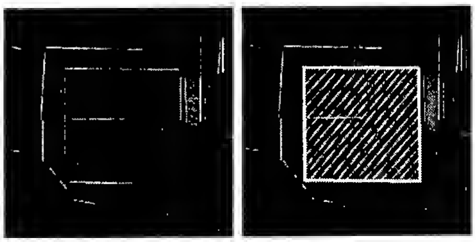
The library of Ascender II algorithms must address aspects of the site reconstruction problem. For example, finding regions that may contain buildings, classifying building rooftop shapes, and determining the position of other cultural features, are all important tasks for the Ascender II system. Many of the IU algorithms may be very "lightweight" and are expected to perform only in a constrained top-down manner. This is due to the fact that the IU algorithms are responsible for gathering evidence for a particular hypothesis put forward by the knowledge base. For example, an algorithm that detects the presence of local maximum in a region of the elevation data can be viewed as a car detector when invoked on a parking lot area. The same algorithm may detect the presence of a rooftop structure when applied to a known building area.

Algorithms may also be very sophisticated, such as the reconstruction of flat roof buildings from multiple views (the role of the Ascender I system). Below, several of the more complex algorithms are briefly described in order to demonstrate the type of algorithms made available to the system.

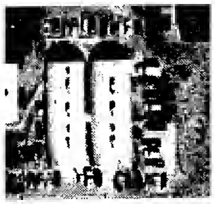
2D Polygon Detection [Jaynes'94]: Search optical image for polygons that represent high confidence rooftop boundaries. Lines and corners are extracted from the image and grouped into perceptually compatible chains. A search of all possible groupings returns the maximal independent set of closed chains.



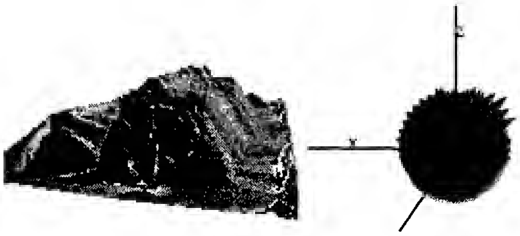
2.5D Feature Grouping: Match image features across multiple images to compute heights and group based on height/shape constraints. For example, compute line heights through a multi-image matching scheme. Group the line segments into sets of two parallel lines at the same height with a third, higher parallel line into regions that may indicate the presence of a peaked roof building.



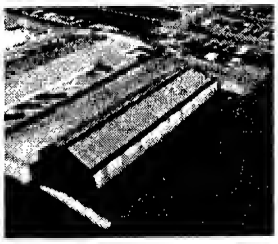
Local Shadow Analysis [Lin et al.'94]: The known sun angle and building model constrains the search for a corresponding shadow in the image. The shape of the shadow can be analyzed to infer the shape of the building rooftop that cast it.



Automatic Model Indexing [Jaynes'97]: Match a region of an elevation image with a surface primitive database. This is accomplished through a construction of the extended Gaussian image for the image region and correlating the surface orientation histogram with the database.



Fitting Parametric Surfaces to DEMs [Jaynes'96b]: Fit a model to a region within the elevation data. The model parameters should have already been determined through another processes (model indexing, for example).



As research into Ascender II continues, more IU algorithms will be added to the system. However, in order for the Ascender II framework to be useful, the cost of adding a new algorithm to the system must not be prohibitive, something that proved to be a problem in earlier knowledge-based vision systems [Draper'89]. Only two components are necessary to convert an IU algorithm into an evidence policy that are usable by the system. First, the context in which the algorithm is intended to be run must be defined. Currently, the definition of allowable contexts is straightforward and only disallows algorithms to be run in invalid contexts (on the wrong type of data, for example). This is similar to the Context Sets introduced in the Condor system [Strat'93] and rule packets within the HUB. This definition of context is expected to be too simple for our needs and eventually the framework will be extended to allow the definition of a performance profile for each algorithm that defines the expected performance of the algorithm under different contexts. Secondly, a method for deriving a certainty value from the output of the algorithm must be defined. This certainty value is used by the system to update the knowledge base using Bayesian inference. For example, the detection of L-junctions within a region of the image must be converted to a single value that represents the probability that the L-junctions are present.

2.3 Preliminary Tests

An experiment was conducted on a scene from the Fort Hood dataset. The test was both a simple example of the concepts presented here and a demonstration of the communication mechanisms that have been constructed as part of the Ascender II system. A small schematic network (only four nodes) was engineered that attempts to classify rectangular building boundaries (called building footprints) according to the three categories, Single, Multi-level, and Multiple that correspond to the case of a single planar rooftop, several planes or slopes at different heights, or more than one building in the region.

The network used for the test is shown in figure 2. The network encodes the fact that the classification of the footprint is dependent upon the presence of certain junction types along the edges of the region, and the quality of a single planar surface fit to the corresponding elevation data. An evidence policy that defined the plane fit algorithm and its reliance on avalable elevation data was constructed for the Plan Level node. Similarly, evidence polices for both L and T junctions were constructed.

Each child node in the network has an associated conditional probability table that encodes object specific knowledge. The conditional probability values are engineered for the specific problem, and, for the test here, were constructed based on our experience with both the evidence policies and the domain.

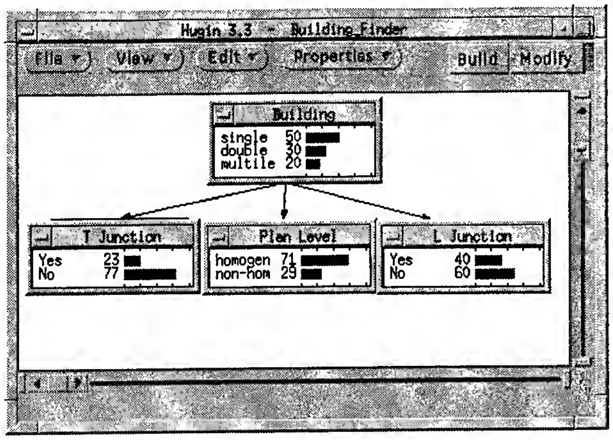


Figure 2: The network used to control the classification of a building region into one of the three possibilities: single building, double building or multilevel building.

The 2D polygon detector used in Ascender I was run on a single downlooking view of a subregion of the Fort Hood dataset. Ten of the polygons were selected for classification using the Ascender II system. In all, four polygons contained single level rooftops, four contained multi-level buildings, and two contained more than one building. Image 3 shows a typical polygon for each of the classes.

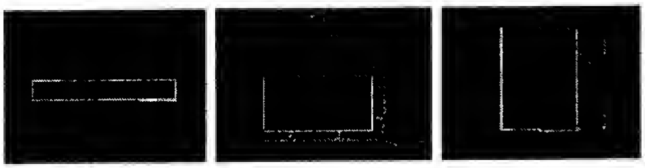


Figure 3: Three different building footprint cases. Leftmost: Single Rooftop. Center: Multi-level Building. Rightmost: Two distinct buildings.

The system was run on all ten regions and stopped when a belief value for one of the states for `footprint class` exceeded 65% or the controller was unable to select an new action. The region is then classified according to the state of `footprint class` with the maximum belief value. The table below shows the results of the experiment and the number of vision algorithms executed in order to classify the region.

Polygon Type	# Actions	Classification/Belief
Single	1	Single (75%)
Mult-Level	6	Multi-Level (57%)
Multiple	4	Multiple (57%)
Single	4	Single (55%)
Multi-Level	6	Multi-Level (50%)
Multi-Level	2	Single (75%)
Single	2	Single (75%)
Multi-Level	4	Multi-Level (83%)
Single	4	Single (81%)
Multiple	6	Multiple (65%)

3 Future Directions

Ascender II is based on a much more flexible design than was Ascender I. Our goal is to demonstrate that this flexibility improves system performance and widens its scope of applicability. To this end, work is underway on engineering the software architecture of Ascender II and on the development of additional evidence policies for a wider range of building classes. The general framework being employed supports any type of data as long as there are corresponding evidence policies available for interpreting it. Consequently, the system is being extended to include IFSAR elevation maps (in addition to elevation maps from traditional stereo techniques) and multi-spectral imagery for improved ground classifications. We expect to use the Fort Hood image dataset as well as other datasets as they become available (e.g. Ft. Benning) to demonstrate the Ascender II system.

There are many issues to be addressed during the design and implementation of Ascender II. One issue concerns the ganularity of the IU algorithms employed in the system and how this affects system performance. For example, should Ascender I be dismantled into component parts and reassembled in the knowledge network? Previous attempts to build knowledge-based systems ran into major knowledge engineering problems. The treatment of IU algorithms as black-box evidence gathering mechanisms, regardless of the underlying complexity, may be one way to avoid this. Currently, simple greedy evidence policy is being used to select the next action. What other policies are reasonable and how do the affect the system efficiency? Techniques that compare the expected utility of applying a particular evidence policy to its expected cost will be investigated as one way to answer the question of efficient control.

References

[Andersen'89] S. Andersen, K. Olesen, F. V. Jensen,F. Jensen, "HUGIN - A shell for building Bayesian belief universes for expert systems" In *Proceedings of the 11th international joint conference on artificial intelligence*, pp 1080-1085, 1989.

[Chellapa et al.'94] R. Chellapa, L. Davis, C. Lin, T. Moore, C. Rodriguez, A. Rosenfeld, X. Zhang, and Q. Zheng. "Site-Model-Based Monitoring of Aerial Images" *Computer Vision and Pattern Recognition (CVPR)*, pp. 694-699, 1997.

[Collins'95] R.Collins, Y.Cheng, C.Jaynes, F.Stolle, X.Wang, A.anson and E.Riseman, "Site Model Acquisition and Extension from Aerial Images," *International Conference on Computer Vision*, Cambridge, MA, June 1995, pp. 888–893.

[Collins, et al'96] R. Collins, C. Jaynes, Y. Cheng, X. Wang, F. Stolle, A. Hanson, E. Riseman. "The ASCENDER System: Automated Site Modelling from Multiple Aerial Images", Submitted to: Special Issue in *Computer Vision and Image Understanding (CVIU)* on Building Detection and Reconstruction from Aerial Images, guest editors R. Nevatia, A. Gruen, to appear 1998.

[Draper'89] B. Draper, R. Collins,J. Brolio, A. Hanson, E. Riseman. "The Schema System", *International Journal of Computer Vision*, vol. 2. pp. 209-250. 1989.

[Gifford and McKeown'94] J. Gifford, D. McKeown. "Automating the Construction of Large-Scale Virtual Worlds *Proc. ARPA Image Understanding Workshop*, 1994.

[Herman'94] M. Herman and T. Kanade. "3D Mosaic Scene Understanding System: Incremental Reconstruction of 3D Scenes from Complex Images". *Proc. ARPA Image Understanding Workshop*, 1994.

[Huertas and Nevatia'80] A. Huertas and R. Nevatia. "Detecting Buildings in Aerial Images" *Computer Vision, Graphics, Image Processing.* vol. 13, 1980.

[Jaynes'94] C. Jaynes, F. Stolle and R. Collins "Task Driven Perceptual Organization for Extraction of Rooftop Polygons," *IEEE Workshop on Applications of Computer Vision,* Sarasota, FL, December 1994, pp. 152–159.

[Jaynes'97] C. Jaynes, E. Riseman, and A. Hanson, "Building Reconstruction from Optical and Range Images" *Computer Vision and Pattern Recognition (CVPR),* San Juan Puerto Rico, June 1997.

[Jaynes'96b] C. Jaynes, F. Stolle, H. Schultz, R. Collins, A. Hanson, and E. Riseman. "Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images" *Proc. ARPA Image Understanding Workshop*, pp. 479-490, 1996.

[Jaynes'96a] C. Jaynes, R. Collins, Y. Wang, F. Stolle, H. Schultz, A. Hanson, and E. Riseman. "Automatic Construction of Three-Dimensional Models of Buildings", chapter to appear in ARPA IU RADIUS book. Oscar Firschein and Thomas Strat (eds.), Morgan Kaufmann Publishers, San Francisco, CA, 1997.

[Jensen'96] F. Jensen, *An Introduction to Bayesian Networks* Springer Verlag New York, 1996.

[Lin et al.'94] C. Lin, A. Huertas, R. Nevatia "Detection of Buildings Using Perceptual Grouping and Shadows", *Computer Vision and Pattern Recognition (CVPR),* pp. 62-69, 1997.

[Matsuyama'85] T. Matsuyama and V. Hwang, "SIGMA: A Framework for Image Understanding: Integration of Bottom-Up and Top-Do wn Processes," *Proceedings of the Ninth IJCAI,* Los Angeles, CA, pp. 908-915, 1985.

[Mundy et al.'92] J. Mundy, R. Welty, L. Quam, T. Strat, B. Bremner, M. Horwedel, D. Hackett, and A. Hoods, "The RADIUS Common Development Environment", *DARPA Image Understanding Workshop*, pp. 215-226, 1992.

[Russel'95] S. Russel and P. Norvig. Artificial Intelligence, A Modern Approach, Prentice Hall: 1995.

[Strat'93] T. Strat. "Employing Contextual Information in Computer Vision", *Proc. ARPA Image Understanding Workshop*, 1993.

Recent advances in 3D reconstruction techniques using aerial images*

Howard Schultz, Frank Stolle, Xiaoguang Wang, Edward M. Riseman, Allen R. Hanson

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
[hschultz, stolle, xwang, riseman, hanson]@cs.umass.edu
http://vis-www.umass.edu/

Abstract

In this paper we focus on two recent improvements in 3D scene interpretation using the UMass terrain and site model reconstruction systems (Terrest and Ascender). The first utilizes a computationally efficient implementation of adaptive windowing to improve rooftop elevation estimates; which are used as input for robust building model generation. The second utilizes intermediate results generated by Terrest to generate a new set of 3D features, which are used as input to a terrain classification system. The techniques were evaluated by analyzing aerial images of Fort Hood, Texas. It was shown that these techniques resulted in significant improvement in the quality of reconstructed building models and the accuracy of terrain classification.

1 Adaptive windowing

The surfac reflectance properties of most man-made and natural objects are approximately Lambertian (i.e., the brightness of a surface feature is independent of the position of observer). Texture based image matching algorithms exploit this phenomena by looking for similar texture patterns in a pair of images. These algorithm have trouble near occluded boundaries, non-Lambertian features, when image noise obscures surface texture, or when the sensors are widely spaced. In previous work, we describe several methods to compensate for perspective distortion introduced by widely spaced cameras (Schultz, 1994, 1995). Although these techniques were shown to be successful in terrain reconstruction, they produced unreliable results near sharp discontinuities in elevation (building edges).

Consider the problem of computing elevations of a flat roof building. When the center of the matching window is located on the roof, and near the corner, most of the windowed pixels are not on the roof. As a result, the ground texture dominates, and the rooftop height is assigned to ground level. This results in a melted building effect, which can be seen on the cover of the 1994 IUW proceedings (Jaynes, et al., 1996).

To prevent errors caused by discontinuities in disparity Kanade and Okutomi (1994) proposed using a deformable matching window. Their method assumes no *a-priori* knowledge about the surface structure. Instead the algorithm assumes that disparity discontinuities are associated with gray-scale discontinuities (a condition commonly found at the edges of building). Although this assumption is valid for many man-made objects, it often fails to describe a wide range of real world conditions, such as rooftop clutter, shadows, patchy vegetation, and road edges. The adaptive windowing technique was adapted to the problem of analyzing complex scenes by employing a building detection algorithm to identify disparity discontinuities.

Instead of looking for jumps in gray-scale to identify disparity discontinuities, we relied on detected rooftop polygons. First, the algorithm detects rooftop polygons in the reference image using a 2D building detection algorithm. The algorithm then generates a

* This work was supported in part by the Defense Advanced Research Projects Agency (via TEC) contract number DACA76-92-C-0041, the Office of Naval Research grant number N00014-89-J-3229, and by the National Science Foundation grant number CDA-8922572.

binary building mask by filling in and dilating the polygons.

When the building mask method is used during reconstruction, only pixels that lie within the mask contribute to the match score. Furthermore, disparities are computed only if the center of the matching window lies within the building mask.

For the reference image, the masking operation is straight forward – all computations affected by the building mask can be carried out in advance. The system pre-multiplies the reference image by the building mask, and counts the number of rooftop pixels surrounding each image pixel. Unfortunately, most of these operations cannot be pre-computed for the target image, because the projection of the building mask onto the target image depends on the elevations, which is not known in advance.

To maintain computational efficiency, the un-masked mean and variance terms in the cross-correlation match score (Schultz, 1994) are substituted for the masked ones. The covariance term is not approximated because all non-roof pixels in the reference image have a value of zero. After the matching process is completed, elevations are generated from the computed disparity map.

The algorithm was tested on a randomly selected cluster of peaked roof building in the Fort Hood data set. Figures 1 shows the reference image, rooftop mask, target image, and the recovered DEMs with and without using the building mask. The fitted peaked roof models generated by Ascender (Jaynes, et al. 1996) are shown in Figure 2. Although the buildings have symmetric peaked roofs, we fitted an asymmetric peaked roof model (the position of the center line was an adjusted parameter). This allowed us to test the effect of the adaptive windowing on the Ascender robust model fitting algorithm. Unfortunately, ground truth is not available for this data set. Nevertheless, based on the position of the roof ridge, it is apparent that the adaptive window improved the quality of the reconstructed model by removing the effects of surrounding objects.

The improvement in the reconstruction accuracy may be attributed to the building mask removing the influence of objects that surround the buildings. The trees at the right and bottom of the scene artificially raise the rooftop elevations, and the parking lot on the left artificially lowers the rooftop elevations.

2 Using 3D features to improve terrain classification

Texture analysis is an important area in computer vision and has been extensively studied (Tamura, et at., 1978; Haralick, 1979; Gool, et al., 1985; and Tuceryan, 1993). While it is widely accepted that texture does not have a precise definition, the importance of texture for terrain classification has been well established. Tuceryan and Jain (1993) identified four basic approaches to texture analysis – statistical, geometrical, model-based, and signal processing methods. Although the methodologies vary substantially from one algorithm to another, they generally assume that texture is a characteristic of the 2D image and indirectly related to the 3D structure of the surface.

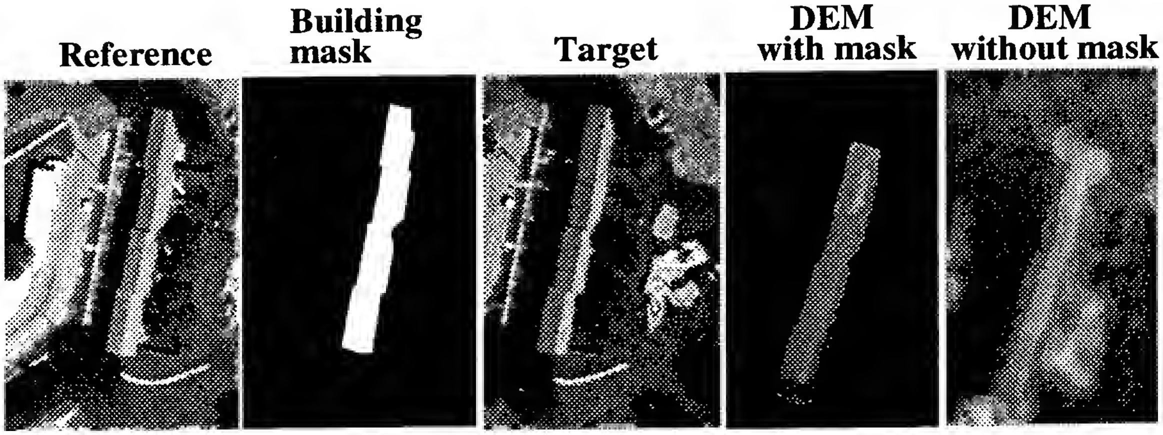


Figure 1. Reconstruction example with and without a building mask.

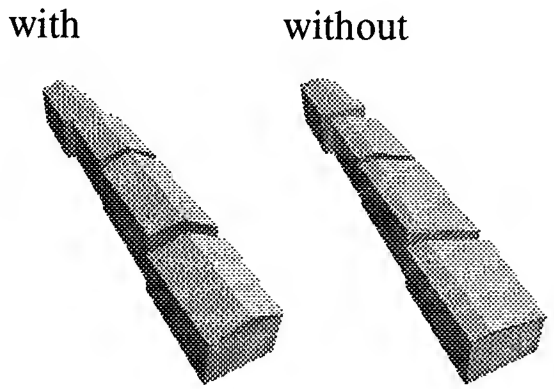


Figure 2. The reconstructed building models.

The limitation of this approach is that the 3D properties of real objects are not utilized. Recently, in a series of experiments at UMass (Wang et al., 1997), we investigated the role of 3D texture on terrain classification. This work examined ways of inferring 3D surface textures (such as coarseness and roughness) from images.

Because information about the 3D micro-structure of a scene usually is not available, 3D features must be derived from images or other remotely sensed data. Instead of reconstructing a scene and then computing a 3D feature set, we propose using the statistical properties of disparity, which are derived from intermediate results computed during the image matching process.

For every pixel in the reference image Terrest searches along epipolar lines in the target image for a match. At all possible disparity, between the upper and lower bound, Terrest generates match scores, which are considered to be independent point estimates of a smooth similarity function (SF).

About its peak the SF is parameterized by a parabola with negative curvature (the peak is the highest point). After computing the series of match scores, a parabola is fitted to the date. If the curvature is negative and the peak lies between the upper and lower bounds of the search range, then the SF is considered well defined.

Four 3D features (which are summarized in Table 1) based on the spatial statistics of the fitted similarity function were computed. Details of the generation of these 3D features can be found in (Wang, 1997a,b).

Experiments were conducted to test the value of 3D features in terrain classification. Figure 3 shows the stereo pair used for the test. The images were 2k × 2k sections extracted from aerial images of a rural area of Fort Hood, TX. The cameras were tilted at 37° and 54°, and the ground sampling distance of the reconstructed DEM was 0.75m. Sample images of the 3D features generated from these data set are shown in Figure 4.

Feature	Description	Characteristics
Peak	Peak value of the similarity function.	High values indicate well defined textures on flat surfaces (rocky ground). Low values are indicative of viewpoint dependent texture, e.g., branches and leaves in a tree canopy.
Variance	Variance of the similarity function peaks in a 17×17 window.	Low values are characteristic of uniform, well defined texture patterns. High values are associated with boundaries and man-made objects.
Curvature	Curvature of the similarity function about its peak.	Large values are associated with high contract textures that vary over a few pixels. Larger values are associated with blurry objects.
Density	Density of well defined similarity functions in a 17×17 window.	Poorly defined similarity functions are associated with areas of unreliable match scores, such as occluded objects and low contrast areas (e.g., paved roads and water).

Table 1. Definitions and characteristics of the 3D features generated from image matching.

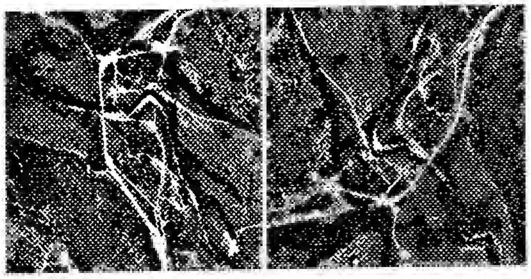


Figure 3. Overlapping pair of images used generate 2D and 3D features.

In addition to the 3D features, a set of twelve standard 2D co-occurrence features (Haralick et al., 1973) also were computed. Three features sets were analyzed: (A) the 12 co-occurrence features, plus image intensities, (B) the 3D features (described in Table 1), plus image intensities, and (C) the combination of A and B.

We selected a linear discriminant classification scheme based on the the Foley-Sammon transform with a minimum Mahalanobis distance selection criterion (Foley, 1975). Four classes were considered in the experiments, namely, foliage, grass, bare ground, and shadow. Very small training data sets were used which accounted for approximately 1% of the total area covered in the orhto-image.

An analysis of the classification accuracy is shown in Table 2. Each row of the table shows the class assignments (in terms of numbers of pixels) for pixels of a known class. The accuracy measure was obtained by summing the diagonal elements and dividing by the total number of pixels. It can be seen that the 2D feature resulted in the lowest classification accuracy, followed by 3D features, and the combined 2D and 3D features. For this experiment, the results show that 3D features significantly improve the classification ability. The classification accuracy of the 3D feature over the conventional 2D set jumps from 72.5 to 82 percent, with only minor improvement when both 2D and 3D feature sets are included.

3 Conclusion

We presented two improvements to 3D scene interpretation and analysis. A computationally efficient adaptive windowing technique was tested in a building reconstruction scenario, and was shown to significantly improve 3D model estimation in complex scenes.

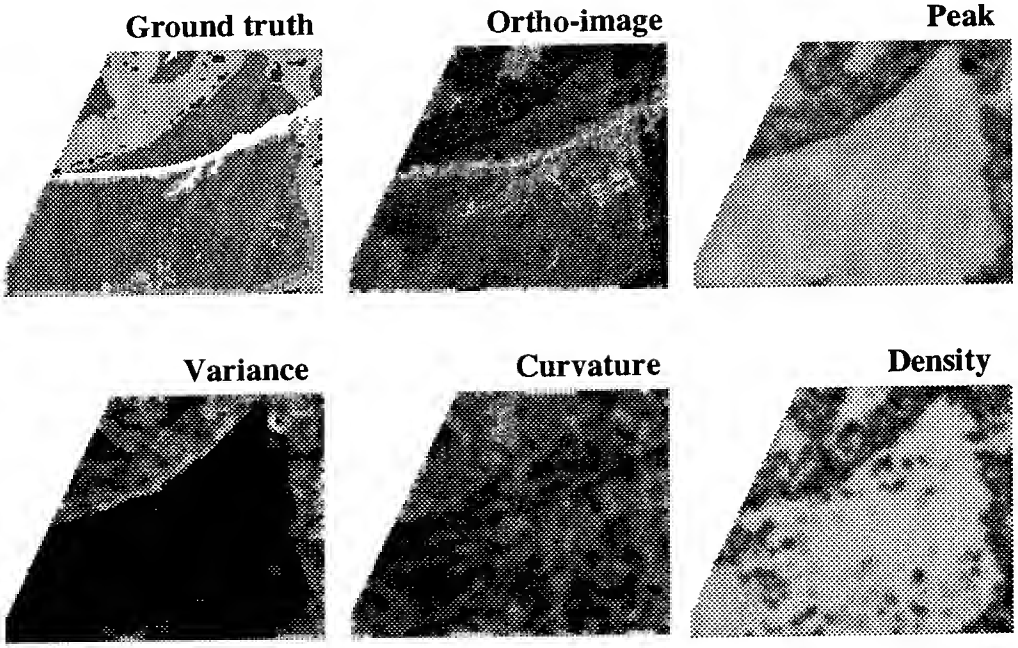


Figure 4. Sample image chips of the ground truth, ortho-image, and 3D features.

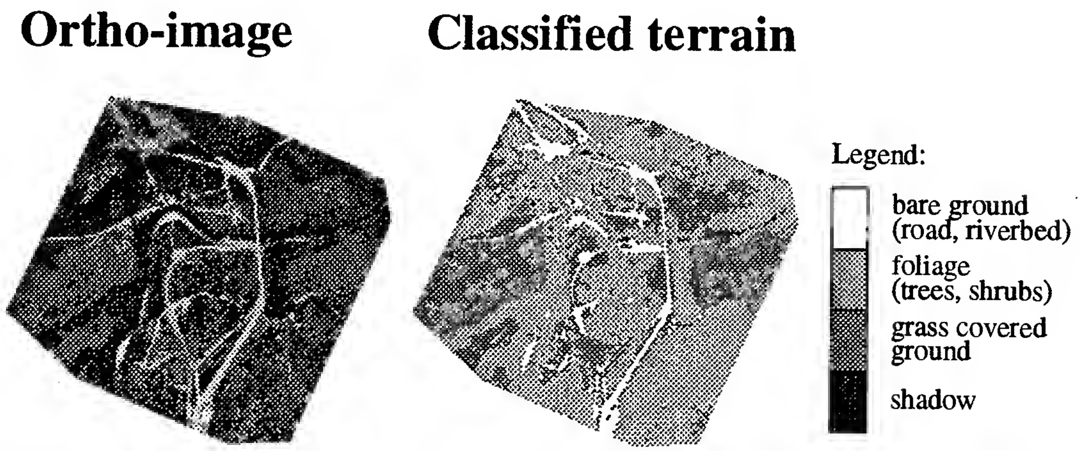


Figure 5. Classification results using 2D and 3D features.

Ground truth	(total)	shadow	grass	foliage	bare ground
2D Features (overall accuracy: 72.5%)					
shadow	(41.8)	**23.7**	7.0	11.1	0.0
grass	(683.0)	40.8	**332.6**	308.1	1.6
foliage	(1018.6)	11.7	55.0	**950.7**	1.2
bare ground	(193.4)	52.2	12.9	31.2	**97.2**
3D Features (overall accuracy: 82.0%)					
shadow	(41.8)	**3.0**	0.0	38.8	0.0
grass	(683.0)	0.0	**439.5**	231.4	12.2
foliage	(1018.6)	0.4	20.0	**995.3**	2.9
bare ground	(193.4)	0.0	17.3	25.1	**150.9**
2D and 3D Features (overall accuracy: 83.4%)					
shadow	(41.8)	**13.8**	0.0	27.8	0.2
grass	(683.0)	0.0	**468.6**	202.7	11.8
foliage	(1018.6)	2.0	18.0	**995.7**	2.8
bare ground	(193.4)	0.0	33.9	21.4	**138.1**

Table 2. Contingency analysis of classification results (units in 1000 pixels).

In the second, we derived a set of 3D features from the intermediate results generated during image matching. The classification ability of 3D features were tested on a rural scene containing four classes (trees, foliage, grass, and bare ground). The results of the experiment showed that using 3D features can significantly improve classification accuracy.

Both of these enhancement have far reaching implications for efficient and robust reconstruction and analysis of complex scenes. In future work these algorithms will be incorporated into the automatic, knowledge driven system Terrest and Ascender, which are currently under development at UMass.

References

D. H. Foley and J. W. Sammon, Jr., "An Optimal Set of Discriminant Vectors," *IEEE Trans. on Computers*, Vol. 24, No. 3, pp. 281-289, 1975.

L. Van Gool, P. Dewaele, and A. Oosterlinck, "Texture Analysis" *Computer Vision, Graphics, and Image Processing*, Vol. 29, pp. 336-357, 1985.

R. M. Haralick, "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE,* Vol. 67, No. 5, pp. 786-804, May 1979.

R. M Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 3, No. 6, pp. 610-621, 1973.

C. O. Jaynes, F. R. Stolle, H. Schultz, R. T. Collins, A. R. Hanson, E. M. Riseman, "Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images," ARPA Image Understanding Workshop, pp. 479-490, Palm Springs, CA, 1996.

T. Kanade, M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment." IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16. No. 9. September 1994.

H. Schultz, "Terrain Reconstruction from oblique views, " ARPA Image Understanding Workshop, pp. 1001-1008. Monterey CA, 1994.

H. Schultz, "Terrain Reconstruction from Widely Spaced Images," Integrating Photogrammetry Techniques with Scene Analysis and Machine Vision II, SPIE Proc., Vol. 2486. Pp. 113-123, Orlando, FL, April 1995.

H. Tamura, S. Mori, and T. Yamawaki, "Textural Features Corresponding to Visual Perception," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 8, pp. 460-473, 1978.

M. Tuceryan and A. K. Jain, "Texture Analysis," In The Handbook of Pattern Recognition and Computer Vision, C. H. Chen, L. F. Pau, and P. S. P. Wang, eds. World Scientific Publishing Co., pp. 235-276, 1993.

X. Wang, F. Stolle, H. Schultz, E. M. Riseman, and A. R. Hanson, "Using Three-Dimensional Features to Improve Terrain Classification," to appear in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

X. Wang, F. Stolle, H. Schultz, E. M. Riseman, and A. R. Hanson, "A New Approach to Terrain Classification Using Three-Dimensional Features," Technical Report, Department of Computer Science, University of Massachusetts at Amherst, March 1997.

Site Modeling Using IFSAR and Electo-Optical Images *

K.B. Hoepfner, Christopher Jaynes, Edward Riseman, Allen Hanson, Howard Schultz
Computer Vision Laboratory, Department of Computer Science
Lederle Graduate Research Center
Box 34610, University of Massachusetts
Amherst, MA 01003-4610
E-MAIL: hoepfner@cs.umass.edu
HOMEPAGE: http://vis-www.cs.umass.edu/

Abstract

The introduction of Interferometric Synthetic Aperture Radar (IFSAR) to the Image Understanding (IU) community provides researchers with a valuable source of elevation data for many common IU tasks including site reconstruction. We show that the new data can either be used to augment processing of existing electro-optical (EO) images of a site or can be exploited as an independent data source. This paper explores the initial application of existing IU techniques to IFSAR images, develops new reconstruction algorithms specifically for noisy IFSAR data, and examines the feasibility of fusing IFSAR and EO algorithms to arrive at a more reliable site reconstruction.

Buildings regions are detected through segmentation of the IFSAR data directly or through delineation of rooftop boundaries in a registered EO image. A top-down model indexing phase automatically determines the rooftop shape of segmented regions. Finally, robust parametric model fitting of the selected rooftop determines the precise shape and location of buildings at the site.

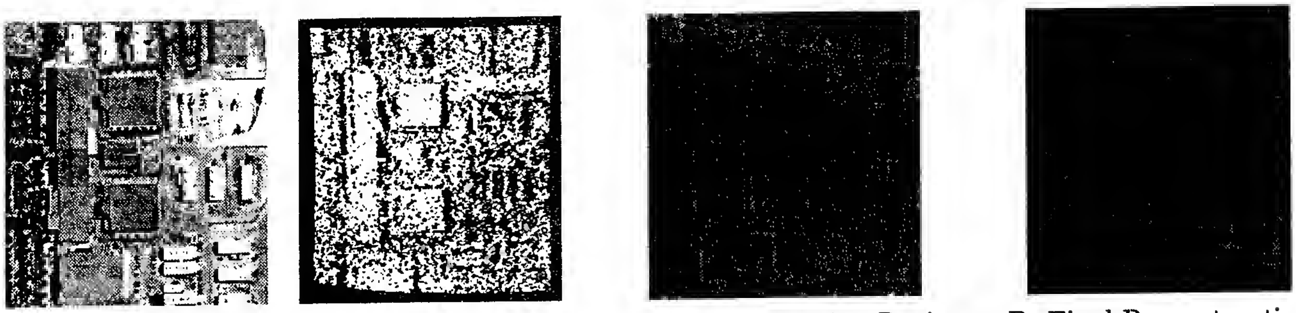
1 Introduction

As part of the UMass effort in the APGD program, there is an ongoing research effort to explore the feasibility of using Inferometric Synthetic Aperture Radar (IFSAR) as a primary data source for IU tasks. In particular, we claim that IFSAR is a valuable source of elevation data that can be used to support 3D site reconstruction tasks [Chellappa et al., 1996]. We have identified some of the characteristics of the data that make the application of IU algorithms to IFSAR a unique problem unto itself. The straightforward application of existing techniques to IFSAR imagery may fail, and numerical methods such as surface fitting may have to be tuned to the statistical properties of the data. Successful exploitation of IFSAR by the IU community will require both the development of new techniques and the strategic application of old ones.

2 A Statistical Profile of the Data

An analysis of the Sandia Kirtland AFB data set was undertaken and two properties of IFSAR relevant to IU algorithms have been noted and focussed upon as part of the study. The first such property is that the Sandra Kirtland AFB data set studied had a large number of dropouts. These are points within the image that contain no data as a result of poor correlations during the reconstruction of IFSAR from its two component synthetic aperture radar (SAR) images [Ulaby et al., 1981]. These dropouts clustered around building edges, often completely occluding rooftop boundaries (see Figure 1B). The second relevant property of the data is a significant presence of outlier noise due to errors inherent

*This work was supported in part by DARPA under contract DACA76-92-C-0041 (via TEC), by NSF grant number CDA-8922572, and by Lockheed Martin under Subcontract/PO number RRM072030.



A: Optical Image B: IFSAR Image C: Focus of Attention Regions D: Final Reconstruction

Figure 1: Figures A and B illustrate the Kirtland AFB data set. Figure C are the rooftop regions found by the technique in Section 3.1 and Figure D is the final reconstruction yielded by applying the model selector to the rooftops in C.

in the IFSAR construction process. Table 1 lists the estimated inlier, outlier and dropout percentages of the Kirtland data set. These estimates were obtained by robustly fitting planes to several rooftops known to be flat and analyzing the residual errors of the fit.

Table 1: Analysis of the Sandra Kirtland AFB dataset.

Inliers	47.87%
Outliers	34.80%
Dropouts	17.33%

The striking frequency of outliers and dropouts has motivated our use of the following IU paradigms in site reconstruction tasks involving IFSAR:

Multisensor Fusion The use of a secondary data type(s) can serve as a means of supporting IFSAR processing in areas where it is weak, while still allowing the exploitation of its strengths. In particular, we focus on using EO along with IFSAR.

Robust Statistical Analysis Robust methods are required in order to overcome outlier noise and converge on correct model fits.

Top-Down, Model Based Analysis The use of building models when applied to the elevation data provides a constraining context helpful in the interpretation of noisy data sets.

It should be understood that the data analysis has been based upon a particular dataset, the Sandia Kirtland data. However, we believe the approaches discussed here will still be valuable and relevant to higher grade data. Note that the reconstruction algorithms presented here were designed with respect to the expected geometry of buildings, not on IFSAR noise-specific artifacts. They are therefore generalizable to other DEM sources of data, such as standard stereo-optical.

3 Methods for Site Reconstruction

The site modeling task focussed upon in this paper is the 3D reconstruction of buildings from an IFSAR image and, possibly, an EO image. The reconstruction process is broken into two separate stages. The first stage locates all the building rooftops in the available data, and three different approaches to this problem were studied. Two of the approaches use a bottom-up, region growing scheme to locate rooftops in an IFSAR image, while the third produces polygonal rooftop boundaries in a registered optical image through the perceptual grouping of image features, such as corners and lines, into closed polygonal forms. The second stage of processing then fits each of these rooftops, or *focus of attention regions*, to a single shape primitive selected from a database of such primitives to yield the final reconstruction.

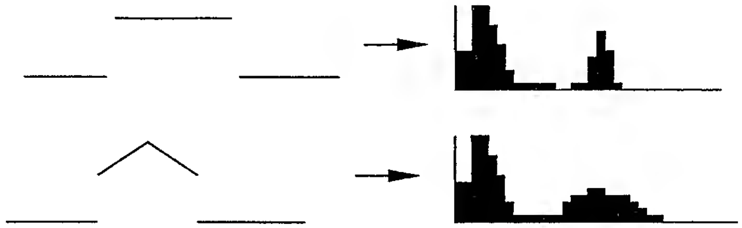


Figure 2: A cross-sectional view of the elevation data for a flat (top) and a peaked (bottom) rooftop and their corresponding projections into an elevation histogram. Note that flat surfaces form sharp spikes in the histogram while peaked surfaces form plateaus.

3.1 Bottom - Up IFSAR processing

The first of the two bottom-up, region growing strategies goes through two separate stages of processing to locate building rooftops in the image. The first stage of processing estimates the likelihood that each point in the image it belongs to some rooftop in the scene. The key spatial constraint is that a building rooftop will appear at a higher elevation than most of the ground immediately surrounding it. If we assume that the entire site lies on a single, flat ground plane whose normal is parallel to the direction of gravity[1], then there are sufficient constraints to allow discrimination between the two via an analysis of an elevation histogram.

One such constraint is that, ideally, every point on the ground will have the same elevation value. Thus, the ground would appear as a unimodal distribution in the elevation histogram of the image. Here, a distribution is considered to be *unimodal* if the number of votes in the bins comprising that distribution decrease monotonically with their distance from the distribution's peak (i.e. the bin with the most votes). Thus, a unimodal distribution is not necessarily a normal distribution in our case. This avoids making strong independence assumptions about the relationship between noise and elevation.

This ground "mode" will be associated with the first significant distribution encountered in the histogram and will be considered to begin at the zeroth elevation bin in the histogram. Furthermore, any distributions formed in the histogram by points lying on building rooftops will be assumed to be distinguishable from, and to the right of, this ground mode (see Figure 2). Rooftop points can therefore be segregated from the ground by thresholding the image at the elevation value corresponding to the valley separating the first (ground) distribution from the other (rooftop) distributions.

The valley best separating the ground mode from other modes in the histogram is selected via a simple fitness heuristic. A local minima's fitness is based on the unimodality of the distribution as defined by that local minima and the magnitude of the division, or bifurcation, created in the histogram by that local minima. A local minima is considered a bifurcation point for a histogram since, ideally, each such minima would represent the bottom of the valley lying between two different modes. These criterion are used to avoid splitting at trivial minima that correspond to artifacts of the image noise, while still selecting a relatively unimodal ground distribution.

The metric for measuring the magnitude of the bifurcation created by a minima naturally suggested by the monotonicity constraint for unimodal distributions is illustrated in Figure 3. It is simply the amount that would have to be "filled in" if the distributions on either side of the minima were to be merged into a single, unimodal distribution. Analysis of the degree of unimodality (i.e. the fitness heuristic motivated by Figure 3) to remove local minima and merge minor modes allows the selection of the key, initial ground plane mode. Points are as-

[1] Note that while this assumption at a global, site-wide level is overly constraining, one can reasonably expect it to hold *locally*, giving rise to the adaptive thresholding approach discussed in Section 4

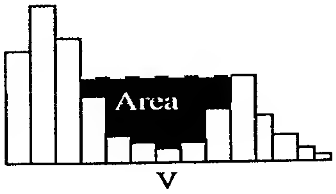


Figure 3: The magnitude for the minima V is simply the shaded area beneath the dashed line. This is the total number of votes that the histogram would have to be changed by in order to accommodate a merger between the modes on either side.

signed likelihood measures based on their distance from the ground mode's peak (i.e. their distance from the ground plane). These likelihood measures are used in the second stage to classify planar patches into roof and ground hypotheses. Next we discus the process of finding the planar patch decomposition.

The second stage decomposes the image into a component set of planar patches and then classifies them into ground or rooftop hypotheses. This segmentation is obtained by growing outward from small seed regions until the error of the best planar fit to that region exceeds an accuracy parameter (this can be related to the expected variance of the data). Seed regions are small blocks of image points to which a single plane can be fit below some residual error. The best planar fit for a region being grown is estimated via a generalized Hough transform [Haralick and Shapiro, 1992] in a three parameter space that describes the surface normal of a plane. This process of locating seed regions and then growing outward from them is iterated until every point in the image belongs to one of these planar regions.

After the image has been fully decomposed into these patches, roof and ground assignments are made to them based on the likelihood measures of their component points (these are the likelihood measures computed in the first stage. Adjacent rooftop patches are then merged whenever such a merger is likely to form a reasonable roof shape (i.e. one that is not concave). The resulting set of rooftop patches are filtered on their size relative to other patches, and the remainder serves as our focus of attention regions. The focus of attention regions located by this technique in the Sandia Kirtland AFB data set are shown in Figure 1C.

The second of the two bottom-up strategies also decomposes an IFSAR image into a set of component planar patches to facilitate rooftop detection, but performs this segmentation in a more sophisticated manner [Piater, 1996]. The image is first decomposed into a set of (possibly overlapping) circular tiles with a user defined diameter. The individual tiles are then fit to planes to create an initial segmentation, after which a split-and-merge phase is entered.

The problem of deriving an optimal splitting and merging schedule for a given tiling is combinatorial in nature and therefore requires the use of an approximate method if a solution is to be tractable. An iterative refinement approach is taken, where, at each iteration, proposed splits and merges are ranked by a locally computed quality heuristic. The merger receiving the highest rank is then instantiated at the end of the iteration, and any other proposal affected by this merger has its quality metric updated. The quality metric for a proposed merge is inversely proportional to the difference between the two candidates' surface normals and the perpendicular distance between them. These iterations are continued until the global error of the planar fits fails to decrease by some pre-specified amount. After the splitting and merging is done, building rooftops are extracted based on elevation.

3.2 Top - Down, Optical Processing

The third approach uses a registered optical image to extract rooftop polygons via a perceptual grouping scheme that delineates rooftop boundaries. The rooftop hypotheses are projected into the registered IFSAR data where they form focus of attention regions. The rooftop classification process described in the next section is then applied to these regions to produce rooftop models. Details on the optical image processing can be found in[Jaynes, 1994].

3.3 Rooftop Shape Classification

Once possible rooftop regions have been extracted, an automatic model indexing phase attempts to classify the their shape. The classification scheme indexes into a database of surface primitives based on an analysis of the differential geometry of the elevation data within each region. Indexing is based on estimating the surface orientations of small surface patches, and constructing an orientation histogram that is then correlated with an existing library of roof models. These orientation histograms, sometimes called the Extended Gaussian Image, are normalized so that they are both scale and translation invariant.

For site reconstruction from IFSAR, the surface primitive database contains a set of surface classes representing possible rooftop shapes called *surface primitives*. Examples are planes, cylindrical surfaces, peaks, and spires. Associated with each surface primitive are a number of models representing different parameterizations of each class of surface primitives. For example, the "Peak" surface primitive class is the canonical shape for a number of models in the database, each with a different peak angle. Corresponding orientation histograms are stored for indexing purposes. Figure 4 shows the database used for the results shown here. It contains 8 different surface primitives and 51 total models.

The elevation data within each region produced by the system is triangulated into a simple surface using the well known Delanuy algorithm. The triangulated surface is a set of oriented triangular surface patches, whose vertices are points from the original data set.

The surface normal at each triangular patch is then computed and used to "vote" for a particular cell on the Gaussian sphere. If the surface normal, N, intersects the Gaussian sphere at (x, y, z), the weighted vote is given by:

$$V(x, y, z, B) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(D^2/\sigma^2)} \qquad (1)$$

where D is the angular distance from (x, y, z) to the center of the histogram bucket, B, to receive the weighted vote. Voting for a given vector stops when the bucket value of $V(x, y, z, B)$ falls below a threshold (0.1 for the results shown here).

To achieve model indexing, the constructed histogram, referred to as the image histogram, is then correlated with each of the model histograms stored in the database. The scheme selected the best model through a comparison of the correlation scores and the orientation at which the model should be fit. The top two models were fit to the elevation data using the parameters derived from the indexing process. The model that has the lowest overall fit error is retained for the final site reconstruction. For a more complete treatment of this algorithm, see [Jaynes,1996]. Figure 1D depicts the reconstruction yielded when the model selector was applied to the focus of attention regions in Figure 1C.

4 Future Directions

Several techniques have been presented that are relevant to the 3D reconstruction of building sites from IFSAR, along with a loose framework for their planned composition into a system. We have also outlined several features desirable in any IU system seeking to utilize IFSAR as a primary data source. One of these is the exploitation of multiple, overlapping sources of evidence when making any judgment about the data. This includes the use of data from multiple sensors and the use of robust statistical techniques.

This project is in the early stages of development, and there are many possible extensions and improvements to the approaches presented here. Data from other sensor types, such as

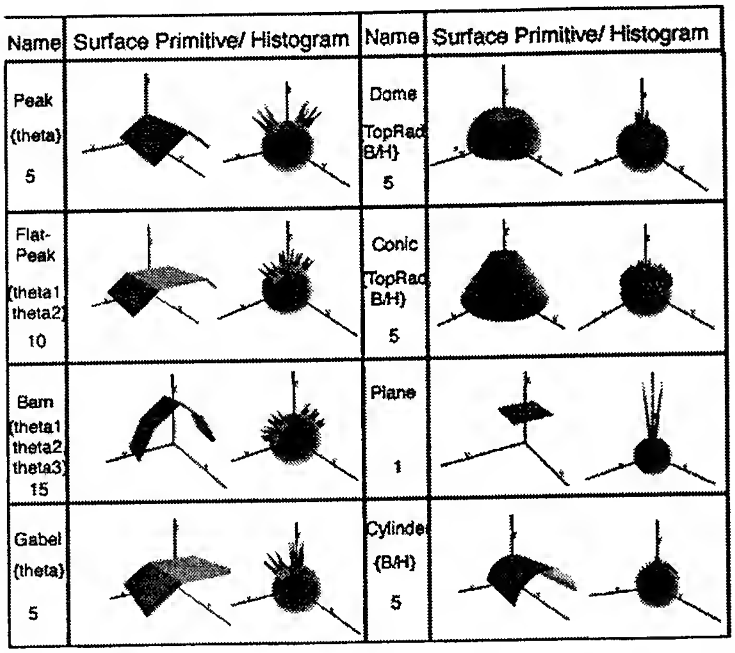


Figure 4: The shape primitive database used in modeling the rooftops detected in the IFSAR image.

SAR or multi-view stereo from optical data may be used to support processing. The use of an adaptive thresholding window in the extraction of rooftop points (Section 3.1) would allow for piecewise - planar ground surfaces, a reasonable assumption when dealing with building sites. The addition of new surface primitives to the model selector's data base (Section 3.3) will be useful, given that such models are separable. A more important improvement is the extension of our modeling capabilities to rooftops of greater complexity through the ability to combine multiple surface primitives in the fit of a single rooftop region. Segmentation in the elevation data would allow a single primitive to be justifiably fit to each subregion which could then be merged into a single, composite model. Such mergings can be accomplished through the use of constructive solid geometry.

References

[Ulabyet al., 1981] Fawwaz T. Ulaby, Richard K. Moore and Adrian K. Fung. *Microwave Remote Sensing: Volume I.* Artech House, 1981.

[Haralick and Shapiro, 1992] Robert M. Haralick and Linda G. Shapiro. *Computer And Robot Vision: Volume I.* Addison-Wesley, 1992.

[Piater, 1996] J. Piater and E. Riseman. Finding Planar Regions in Noisy 3D Grid Point Data. Computer Science Technical Report 96-47, University of Massachusetts, Amherst (MA), 1996.

[Jaynes, 1994] C. Jaynes, F. Stolle and R. Collins. Task Driven Perceptual Organization for Extraction of Rooftop Polygons. *IEEE Workshop on Applications of Computer Vision*, pp. 152–159, Dec 1994.

[Chellappa *et al.*, 1996] R. Chellappa, S. Kuttikkad, R. Meth, P. Burlina, K. Ome, C. Shekkar Model Supported Exploitation of SAR Imagery *Proc. ARPA Image Understanding Workshop*, pp. 389–408, 1996.

[Jaynes, 97] C. Jaynes, E. Riseman, and A. Hanson. Building Reconstruction from Optical and Range Images. *Proc. ARPA Image Understanding Workshop*, pp. 479–490, 1996.

Detection and Description of Buildings from Multiple Aerial Images

Sanjay Noronha and Ram Nevatia*
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089-0273

Abstract

A method for detection and description of rectangular buildings from two or more registered aerial intensity images is proposed. The output is a 3D description of the buildings, with an associated confidence measure for each building. Hierarchical perceptual grouping and matching across views is employed to increase the robustness of the system. Verification of selected building hypothesxses is done using shadow and wall evidence of the buildings. The system is largely feature-based. Grouping and matching are performed in a hierarchical manner, utilizing primitives of increasing complexity, starting with line segments and junctions, and proceeding to higher level features. Binocular and trinocular epipolar constraints are used to reduce the search space for matching features.

1 Introduction

Detection and description of building structures from aerial images is becoming increasingly important for a number of applications such as map-making, change detection and databases for simulators. This problem also offers an opportunity of exploring the issues of object segmentation and 3-D shape inference in a limited setting but where significant challenges must be met. While this problem has been approached with use of just a single image ([5], [7]), multiple images of the same scene are often available. In this paper, we assume that two or more images are available though they may not be taken at the same time and so the imaging conditions may be quite different.

The task of detecting and describing buildings presents many challenges. In a single image, the object boundaries are typically highly fragmented due to low contrast, occlusion caused by nearby vegetation and by smaller structures on the roofs, and need to be grouped to yield the desired objects. In our work, we limit the buildings shapes to be rectilinear (i.e. rectangular or compositions of rectangular shapes) to aid the task of organization. However, many other structures such as roads, sidewalks and parking lots can also give rise to rectilinear organizations and need to be distinguished from the building structures. Availability of multiple images allows the possibility of doing some of this reasoning in 3-D, by making correspondences between the image features. This task too is difficult in the aerial image domain. Area correlation methods are likely to have difficulty as the viewpoints can be widely separated, the images are taken at different times and the building roofs have limited texture. In our system, we choose to match features instead.

Figure 1 shows two views of a scene. Figure 2 shows the line segments detected in the images in Figure 1. These views illustrate some of the difficulties that arise. A large number of line segments are detected but only a few correspond to boundaries of desired structures. Many of the lines are parallel to each other and hence difficult to match in the two views unambiguously, without higher level context. Also, many rectangular organizations of the features are possible if fragmentation is allowed. In addition, poor camera calibration prevents us from making highly accurate 3D position inferences, which complicate the task of higher level segmentation and description.

An important question in multiple image analysis is the level at which image features should be corresponded. Lower level features, such as edges, are easy to detect but are highly ambiguous. Higher level features, such as surfaces, are easily matched but hard to detect reliably in single images. Some systems have been constructed to match features such as junctions ([1], [10]) which are then used for grouping. Other systems have attempted to find candidates for roof boundaries and match them or verify them to get 3-D descriptions ([8], [2]). We feel that matching at only one level does not fully exploit the information available in the multiple images and that rather than deciding between grouping first and then matching, or matching first and then grouping, it is more advantageous to interleave the two processes so that local features are matched and then grouped to form higher level features in a hierarchical way. While hierarchical approaches have been suggested in the past (for example, [4], [6]) they have rarely been implemented for scenes of complexity considered here.

A block diagram of our approach is shown in Figure 3.

* This research was supported in part by Contract No. 76-93-C-0014 from the Defense Advanced Research Projects Agency (DARPA) and monitored by the Topographic Engineering Research Center of the US Army, and by the US Air Force Office of Scientific Research.

Our approach is to first form hypotheses for building roofs as roofs project into larger areas and to verify the hypotheses by using evidence from cast shadows and visible walls (if any). As we consider only rectilinear buildings, a natural hierarchy for hypotheses formation is that of lines, junctions, parallels, Us (three sides) and parallelograms (roofs project into parallelograms since the imaging distance is large compared to the height of the buildings). Matching at one level is used to form group hypotheses at the next level. We maintain multiple matches at each level and resolve them only when sufficient information becomes available at the higher levels. We believe that this approach not only provides good results for the building detection task but also provides a model for more general conditions.

Our system has been tested on a number of real images. The details of our system and some results are shown in the following sections.

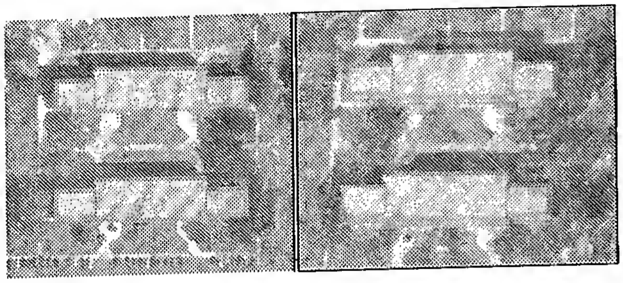


Figure 1 Two views of a scene

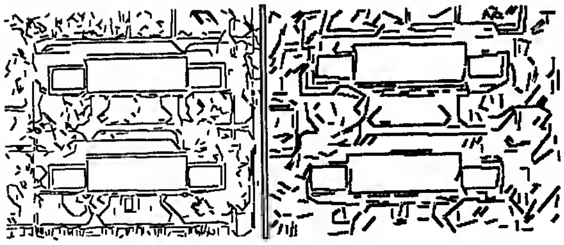


Figure 2 Lines detected in Figure 1

2 Hierarchical Grouping and Matching of Features

The system is built to handle two or more views non-preferentially. A hierarchy of features is used. Starting from the most primitive these are lines, junctions, parallels, U's and parallelograms (because the projection of a rectangle is a parallelogram in general, assuming negligible perspective effects), in each image. Grouping and matching is performed at each stage. Below we explain which features in the hierarchy were chosen for matching purposes.

2.1 Lines

Lines detected using the Canny edge-detector are matched across all views. Multiple matches are retained at this stage. The constraint used in matching is the quadrilateral constraint described in [9].

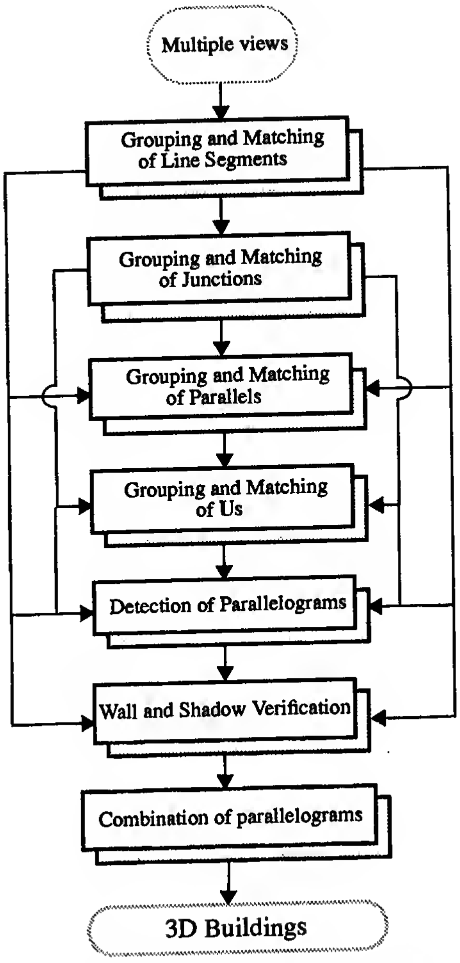

Multiple views
Grouping and Matching of Line Segments
Grouping and Matching of Junctions
Grouping and Matching of Parallels
Grouping and Matching of Us
Detection of Parallelograms
Wall and Shadow Verification
Combination of parallelograms
3D Buildings

Figure 3 Block Diagram of the System

2.2 Junctions

Next, binary junctions (formed by the intersection of exactly two lines) are formed and matched. The constraints for junction matching are the epipolar constraint, the line-match constraint, the 3D Orthogonality constraint, the 3D height constraint and the trinocular constraint. These are outlined in [9].

2.3 Parallels

Next, we search for parallels and their matches. Parallels are formed between pairs of lines in the same view that are separated by the less than the maximum width of a building. A match is hypothesized if there is evidence in at least two views. The parallel match constraint described in [9] is used to remove parallel matches that cannot lead building hypotheses that are planar in 3D.

2.4 U's

U's are formed by an alignment of two junctions, or by a parallel that has closure evidence near one of its ends. The presence of a U is a strong indication of a fragment of a parallelogram in 2D (implying a possible rectangular fragment in 3D). [9] contains details of the implementation, and the constraint applied at this stage.

2.5 Parallelograms

Formation of parallelograms is the basis for hypothesizing buildings. The existence of evidence to form a parallelogram match, is a strong indication that a rectangular 3D structure exists.

3 Selection of Building Hypotheses

The parallelogram matches serve as roof hypotheses, and are equivalent to having a 3D model of the buildings. Owing to the resolution of the images, and the large errors in triangulation from small errors in the images, additional processing needs to be done to distinguish which hypotheses are buildings or parts thereof, and which are rectangular areas on the ground. This necessitates a selection procedure. The selection procedure uses four criteria to decide which hypotheses should remain for verification, namely the 3D height of the building, positive and negative line evidence, and junction evidence. Details are given in [9]. The results after the selection procedure are shown in Figure 4

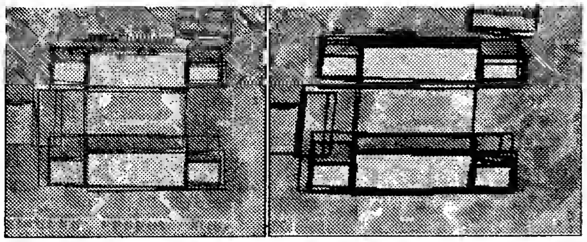


Figure 4 Selected hypotheses from Figure 1

4 Verification of Building Hypotheses

It may be noted that so far the evidence that was used was concerning the roof only. The presence of lighting causes shadows to be cast. When the view is oblique, some vertical sides of the walls of the building may be visible. These cues are used to verify the selected hypothesis, and further reduce the number of hypotheses based on the available evidence. The numerical evidence for the walls and shadows is accumulated for all the views, and the average is compared against a threshold. These monocular cues are extremely important when the registration has errors large enough to cause height estimates to be unreliable. This is the case with the Fort Hood images.

Wall evidence. In a view which is not nadir one or more of the side walls of the buildings should be visible. These walls are assumed to be vertical. The details of verification for walls are included in [9].

Shadow evidence: When shadow lines are present, they are used to boost the confidence of the hypothesis. In case of the Fort Hood images, shadow evidence is often a major factor in the verification of a selected hypothesis.

The evidence of shadows and walls is combined in a Bayesian manner, with a priori probability estimates obtained from the expected length of the shadow (wall) line).

The combined score from the wall evidence and shadow evidence is thresholded to obtain rectangular building (or building component, in the case of non-rectangular rectilinear buildings) hypotheses.

5 Combination of Rectangular Buildings

Rectilinear buildings can be decomposed into rectangular components. Verified rectangular hypotheses are examined for combination according to two mutually exclusive criteria: proximity, and overlap. The precondition for both criteria is that the hypotheses be of approximately the same height in 3D.

6 Results

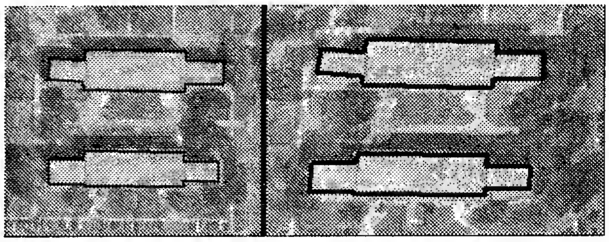


Figure 5 Results after combination from Figure 1

We have used images from Ft. Hood, Texas, for our experiments.This dataset was acquired for the U.S. government sponsored RADIUS program and has become a common "benchmark" for evaluation of building detection systems. This site is challenging, because it has non-rectangular buildings, vehicles are present on the roads and parking lots, and it has trees and grassy areas. Real lighting conditions cause shadows that are not necessarily the darkest areas in the images. Furthermore the acquisition geometry is such that the epipolar lines between many pairs of views are almost parallel (within 5^{o}) to one of the sides of the buildings (in at least one view) at the site. This causes height estimates to be less reliable and the selection process less certain.

Figure 5 shows the results obtained from the images in

Figure 1. Both buildings in this example are described. The example indicates how the combination routine may be recursively applied to combine rectangular building components into a rectilinear building.

Figure 6 shows L-shaped buildings, with markings on the roofs. It may be noted that the structural features are parallel, giving rise to a number of parallel structures that the program has to disambiguate between. Figure 7 shows T-shaped buildings. There is fairly dense vegetation close to the buildings, causing shadow evidence to be obfuscated in some areas. The small sections of the building near the top edge of the image are not verified because large sections are occluded by the shadows of the long parts of the building. Figure 8 shows the results obtained for fairly complex I-shaped buildings. It may be noted that the view depicted is fairly oblique. There is one false positive in this example. This is caused by accidental alignment of shadows and the side of the building. Figure 9 shows a number of small, low buildings. The height estimates for these buildings are extremely unreliable (the buildings are less than 5m high). Shadows and walls are extremely important in this case. Some of the small buildings were not verified because there was not enough edge evidence to support their selection. Figure 10 shows the performance on extremely complex A-shaped buildings. Many shadow and wall areas of rectangular components of these buildings are occluded by other parts of the buildings. This example proves that the system can work when significant amounts of evidence do not exist. Figure 11 shows two buildings with wings. This illustrates the working of the system with buildings with few features on the roof (which might create problems for area-based systems). Figure 12 shows the results on some multi-level and gabled-roof buildings. The multi-level building is not perceived as a multi-level building, because the difference in heights of the levels is too small to be detected in the presence of the registration errors. Figure 13 shows results on a fairly complex area. Most of the buildings are detected and modeled correctly, inspite of occlusion from vegetation and poor shadows. Some errors or deficiencies can also be seen. Some components of multi-wing buildings are not detected because of missing line evidence, such as the component labeled **A**. The building labeled **B** is an instance of a "true negative". This building is missed because of occlusion by shadows of the neighboring building, and the low height of the building itself.

We have processed large areas of the "Motor Pool Area" of the Fort Hood images as shown in Figures 14, 15 and 16. Figures 15 and 16 are reproduced at low resolution to show the large sections. Figure 14 shows a sub-section at the resolution at which data is processed. These results were obtained by using the depicted view with one other overlapping view. There are a number of multi-wing buildings, flanked by smaller rectangular buildings. The rooftops of these buildings are very similar photometrically, to the ground. None of these buildings is taller than 15m. Inspite of these difficulties, the system reliably finds the large buildings in areas where the sides of the buildings are not highly fragmented owing to the similar reflectance properties of the buildings and the ground near it. It performs less reliably when the epipolar lines are parallel to the sides of the buildings as matching these lines is harder than when the lines form a significant angle with the epipolar lines. For example, the building labeled **C**, in Figure 14 is inaccurately modeled. This error is caused by accidental background geometrical formations. Better registration would permit higher confidence 3D height estimates, facilitating better selection.

Evaluation of the system is performed using quantitative and qualitative criteria. A model is constructed by hand for comparison. A building is declared detected if its roof area overlaps more than 50% of a roof of a building in the supplied model. Quantitative measures of the performance of the system may be defined as follows: if t_p is the number of true positive hypotheses, t_n is the number of true negative hypotheses and f_p is the number of false positive hypotheses, then we define the detection percentage as $t_p/(t_p + t_n)$, and the branching factor as $f_p/(t_p + f_p)$. For one part of the site from the Motor Pool Area of Fort Hood, TX, (shown in Figure 15), t_p was 51, t_n was 11, and f_p was 5. For another part of the site, from the Motor Pool Area (shown in Figure 16) t_p was 25, t_n was 7 and f_p was 4. Measures of the number of pixels that are correctly labeled as building and non-building pixels are also useful. They are obtained by comparison with the supplied model.These measures are shown in **Table 1**.

We are unable to compare our results with those of other researchers directly as we do not have access to their software. We can compare to their previously published results, however, they may not be on the same data even when they may have used the Ft. Hood data set and the published results are necessarily outdated.

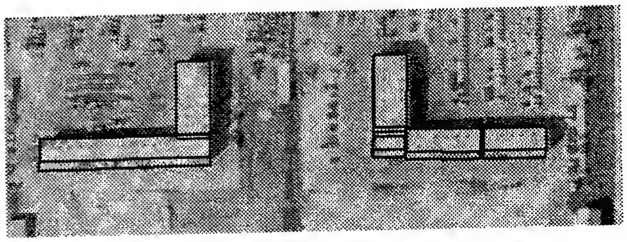


Figure 6 L-shaped buildings

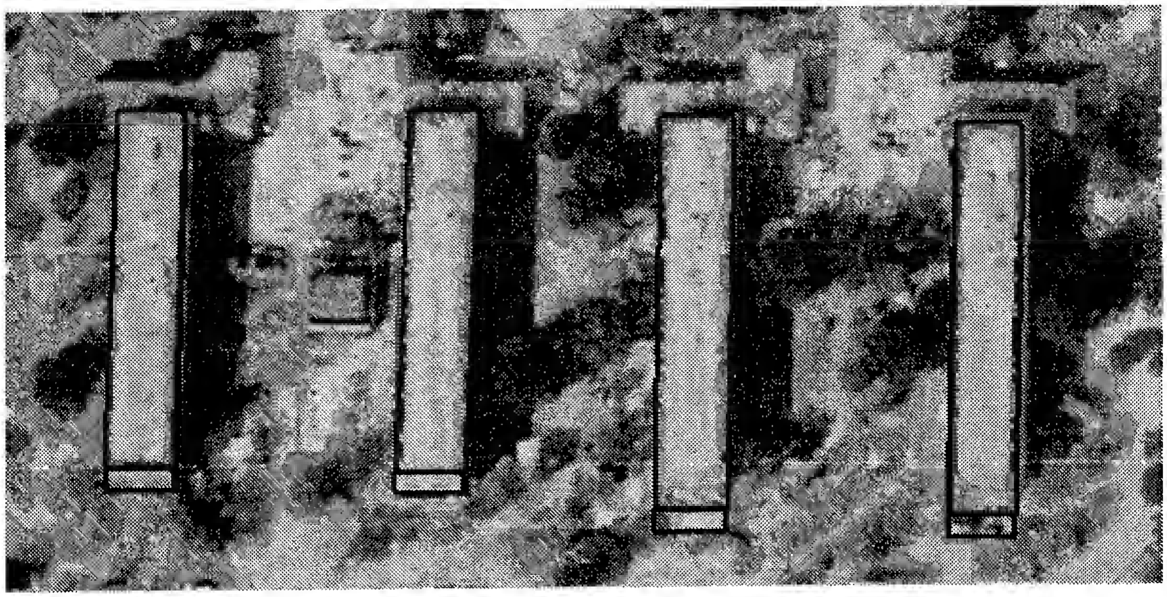


Figure 7 T-shaped buildings

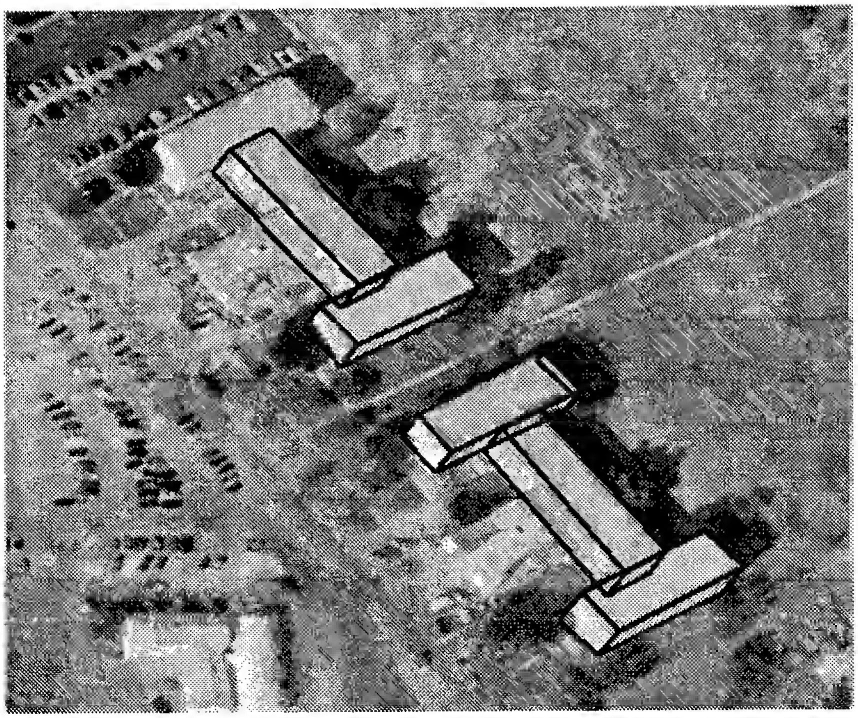


Figure 8 I-shaped buildings

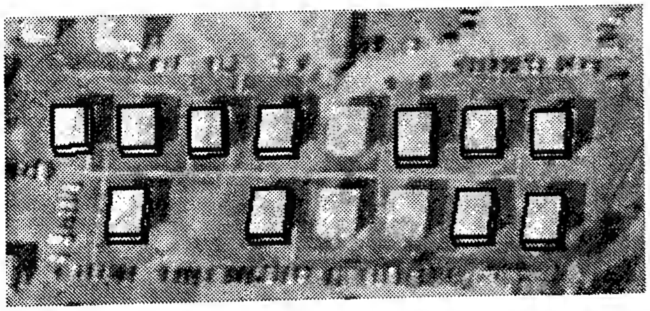


Figure 9 Rows of small buildings

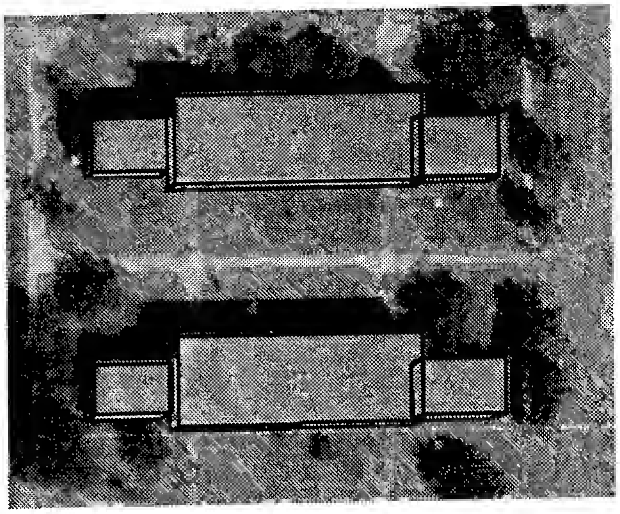


Figure 11 Buildings with wings

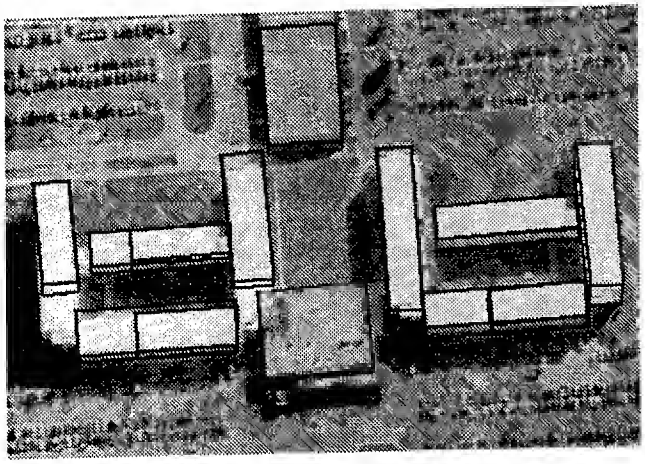


Figure 10 A-shaped buildings

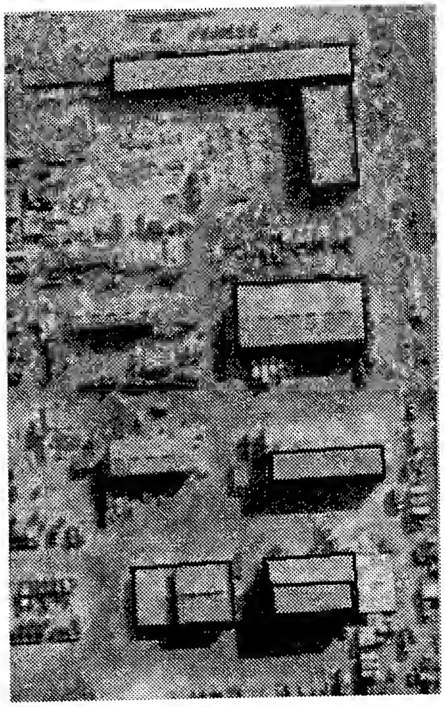


Figure 12 Gabled roof and multi-level buildings

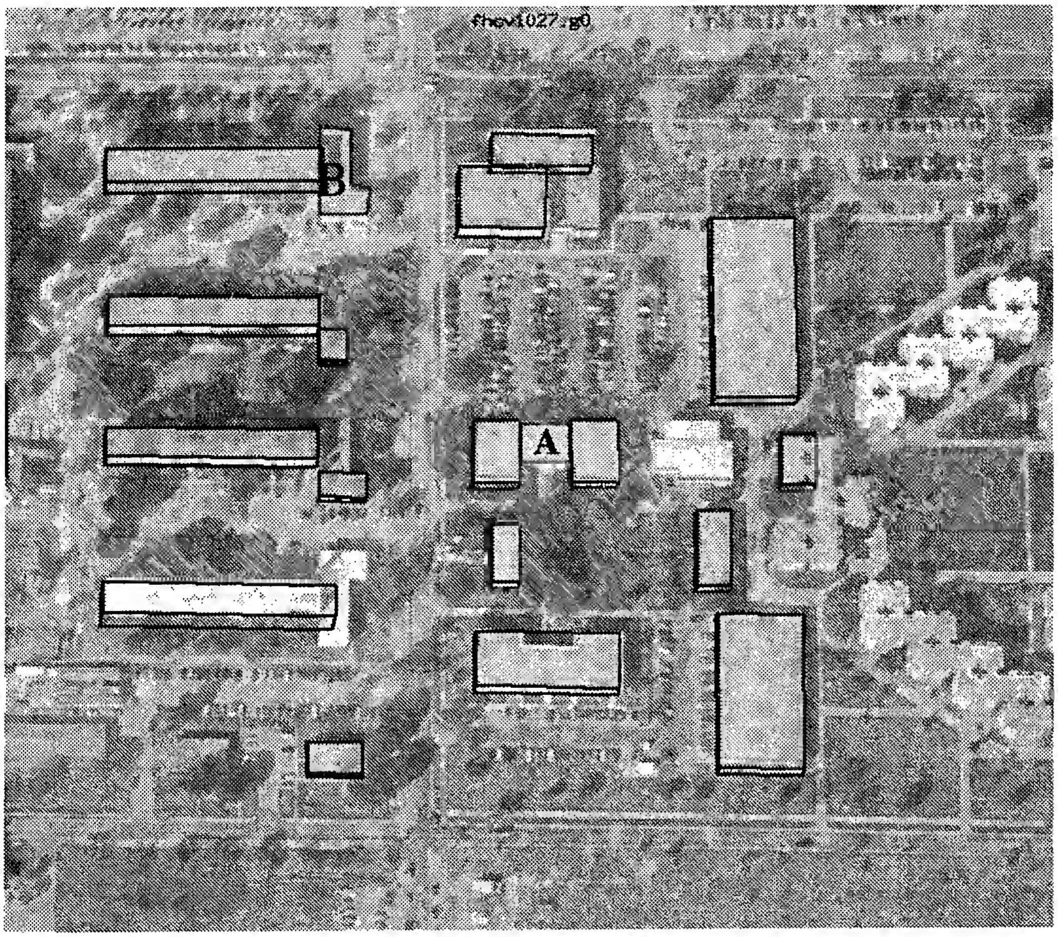


Figure 13 Results on a complex area of Fort Hood

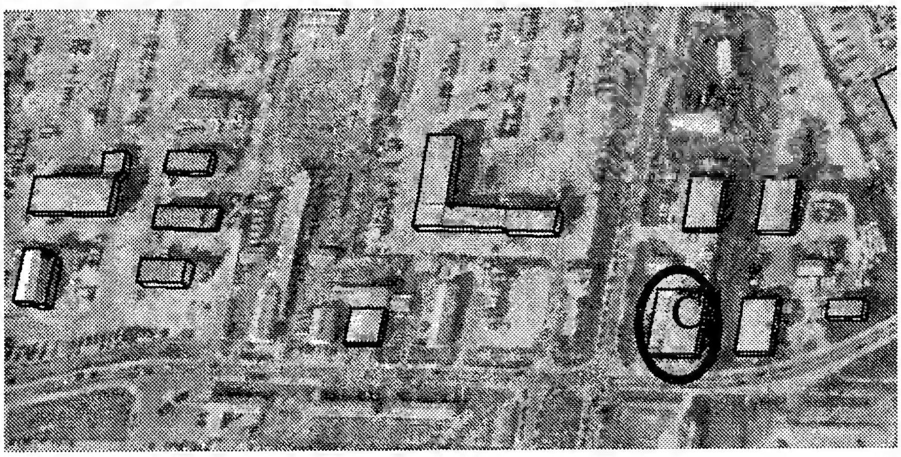


Figure 14 Section of the Motor Pool Area of Fort Hood, TX

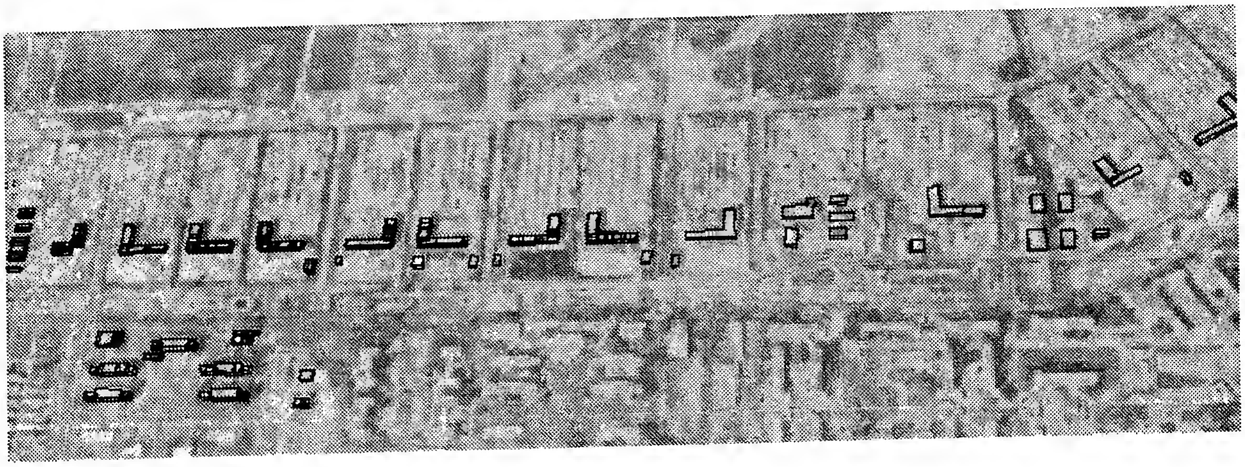


Figure 15 Section 1 of the Motor Pool Area from Fort Hood, TX

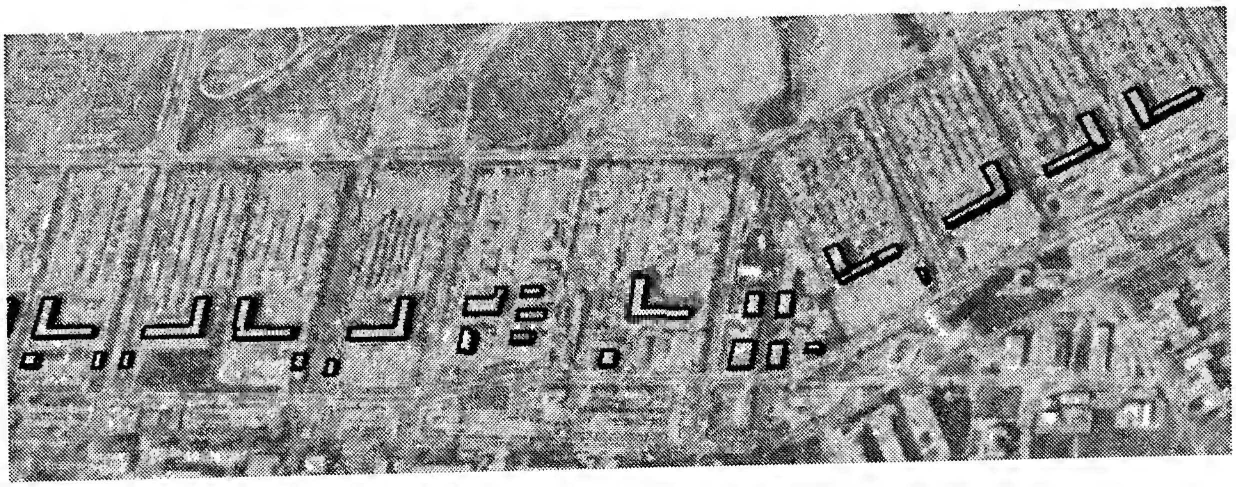


Figure 16 Section 2 of the Motor Pool Area from Fort Hood, TX

Table 1:

Section	Detection Percentage $t_p/(t_p + t_n)$	Branching Factor $f_p/(t_p + f_p)$	Correct building pixels	Correct non-building pixels
Section 1	82.26%	0.08929	75.36%	99.13%
Section 2	78.13%	0.13333	71.84%	98.72%

We feel that the results we show, particularly in Figure 13, are on much more complex examples and that we have processed much larger number of buildings than reported in the previous literature ([5], [2], and [10]). We feel that our method has advantage over monocular systems such as ([5]) due to its ability to infer 3-D more directly and of having several views to validate the hypotheses. We feel that our system can perform more consistently over systems such as ([2]) which form hypotheses in one view and verify in others by not being dependent on a favored view.

Bibliography

[1] R.C.K. Chung and R. Nevatia, "Recovering building structures from Stereo", IEEE Proceedings of Workshop on Applications of Computer Vision, 64-73, Dec 1992.

[2] R. Collins, Y. Cheng, C. Jaynes, F. Stolle and X. Wang, "Task Driven Perceptual Organization for Extraction of Rooftop Polygons", Proceedings of International Conference on Computer Vision, 6:888-893, June 1995.

[3] M. Ito and A. Ishii, "Three-view stereo analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:524-532, 1986.

[4] H.S. Lim and T.O. Binford, "Stereo correspondence: A hierarchical approach", Proceedings of DARPA Image Understanding Workshop, 234-241, 1987.

[5] C. Lin, and R. Nevatia, "3D Descriptions of Buildings from an Oblique View Aerial Image", IEEE International Symposium of Computer Vision, 377-382, 1995.

[6] X.C. Magnisalis and K.L. Boyer, "Hierarchical structural stereo matching with simultaneous autonomous camera calibration", Proceedings of International Conference on Pattern Recognition, 711-713, 1994.

[7] J. McGlone and J. Shufelt, "Projective and Object Space Geometry for Monocular Building Extraction", IEEE Proceedings of Computer Vision and Pattern Recognition, 54-61, 1994.

[8] R. Mohan and R. Nevatia, "Using perceptual organization to extract 3-D structures", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(11), 1121-1139, Nov 1989.

[9] S. Noronha., R. Nevatia., "Detection and Description of Buildings from Multiple Aerial Images", Proceedings of Image Understanding Workshop 1996, Palm Springs, pp. 469-478.

[10] M. Roux and D.M. McKeown, "Feature matching for building extraction from multiple views", IEEE Proceedings of Computer Vision and Pattern Recognition, 46-53, 1994.

Ridge and Ravine Detection in Digital Images

Thomas C. Henderson, Scott Morris, and Charlotte Sanders
Department of Computer Science
University of Utah

Abstract

A novel and efficient ridge and ravine detection method is given.

1 Introduction

Ridges and ravines are important features in some image analysis tasks and represent a basic topographic type in digital terrain data. Several methods have been proposed to recover these features, but they have major shortcomings including (1) their sensitivity, and (2) their computational cost (usually as a result of fitting a polynomial). We describe here an approach based on the Laplacian operator that has a firm theoretical foundation and which is relatively inexpensive to compute.

Haralick[Haralick and Shapiro, 1992] describes how the facet model approach can be used to recover ridges and ravines. A bicubic polynomial is fit to a patch in the image; ridges are then characterized by a negative second derivative across the ridge line and a zero first derivative in the same direction. The only difference for a ravine is that the second derivative across the ravine is positive. (Haralick's book reviews several earlier techniques for ridge and ravine detection; note that Rosenfeld and Kak maintain that the Laplacian can be used to detect lines.) The computational cost is high due to the ten coefficients that are computed at each pixel.

A more recent technique related to our approach is that proposed by Gauch and Pizer[Gauch and Pizer, 1993]. In their approach, they find places where the "intensity falls off sharply in two opposite directions." They determine curvature extrema of the level curves of the image in order to achieve this. Unfortunately, their calculation requires the evaluation of a large polynomial in the first-, second- and third-order partial derivatives of the image, where cubic splines are used to calculate the partial derivatives.

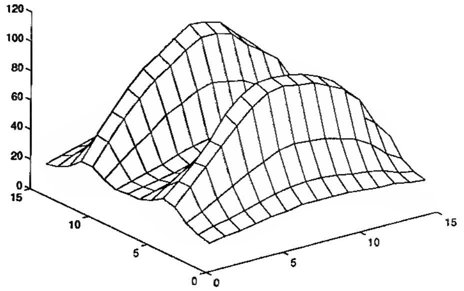


Figure 1: Ridges Viewed as Topography

2 Curl Method

Our method is based on the following sequence of observations concerning the behavior of the gradient in the neighborhood of a ridge or ravine. Figure 1 shows an image as a surface in 3D (this is a subimage of a medical image with intensity bands). The gradient (see Figure 2) produces vectors on the side of a ridge which point toward the ridge and which point away from a ravine. Although the gradient can be analyzed directly to determine the location of ridges and ravines, it is computationally more convenient to do the following:

- Rotate (locally) each gradient vector -90 de-

This work was supported by the Defense Advanced Research Projects Agency, contract number pending.

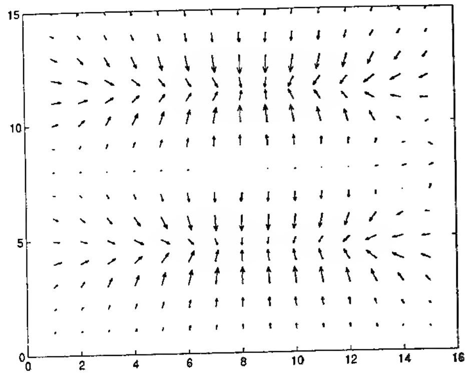


Figure 2: Gradient Vectors of Ridge Image

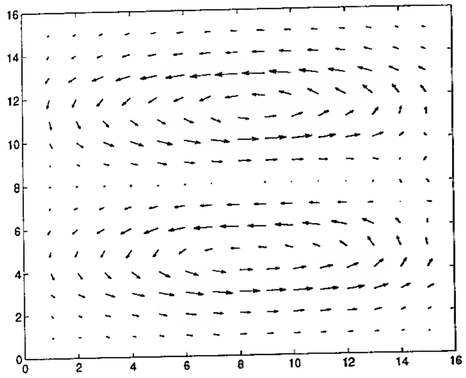


Figure 3: Rotated Gradient

grees about the out of image axis.

- Calculate the curl at each point to determine the opposed flow that exists at ridge lines.
- Calculate the extremum of this function across the ridge.

Figure 3 shows the rotated gradient for the image of Figure 1, while Figure 4 shows the extracted ridge pixels.

Now that the ideas should be clear, we give a formal development of this technique. Let the image function be $f(x,y)$. Then the gradient is:

$$\bigtriangledown f = f_x(x,y)\cdot\bar{i}+f_y(x,y)\cdot\bar{j}+0\cdot\bar{k}$$

The rotation is:

$$rot(\bigtriangledown f) = f_y(x,y)\cdot\bar{i}-f_x(x,y)\cdot\bar{j}+0\cdot\bar{k}$$

The curl of this is:

$$curl(rot(\bigtriangledown f)) = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ f_y & -f_x & 0 \end{vmatrix}$$

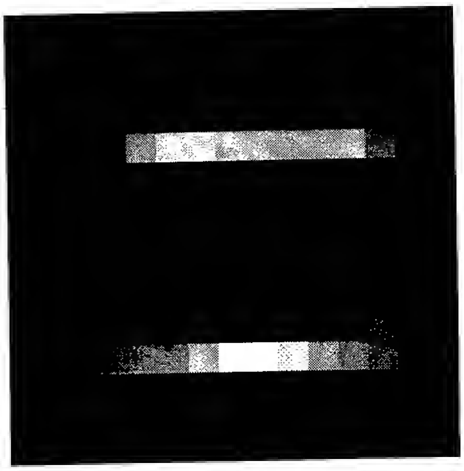


Figure 4: Extrema of Curl across Ridge

$$= 0\cdot\bar{i}+0\cdot\bar{j}+(-f_{xx}-f_{yy})\cdot\bar{k}$$

which is just the negative of the Laplacian.

Finally, a principal direction of curvature for a ridge pixel is:

$$\alpha = \frac{atan2(f_{xy}, f_{yy}-f_{xx})}{2}$$

as well as $\alpha+\frac{\pi}{2}$. We search in these directions to determine that the pixel is a local maximum across the ridge. Note that ravines can be found in a similar way, as they are (negative) minima. (Also, it is possible that vectors of different strengths pointing the same direction give rise to a response; they can be easily filtered if necessary.)

3 Summary

We believe that the curl of the rotated gradient provides an excellent basis upon which to construct a robust ridge and ravine detector. It is comparable to existing operators, but much less costly. We have tried this technique on a number of types of images and found the results to be very good.

References

[Gauch and Pizer, 1993] John Gauch and Stephen Pizer. Multi-resolution analysis of ridges and valleys in grey-scale images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):635–646, June 1993.

[Haralick and Shapiro, 1992] Robert Haralick and Linda Shapiro. *Computer and Robot Vision.* Addison-Wesley Pub Co, Reading, MA, 1992.

Extraction of Micro-Terrain Ravines Using Image Understanding Constrained by Topographic Context

Gregory W. Thoenen and **William B. Thompson**
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
http://www.cs.utah.edu/projects/robot/micro-terrain/

Abstract

Current methods for generating terrain databases do a poor job of integrating drainage features into the terrain skin, and in arid areas leave out features such as ravines and dry washes altogether. This paper demonstrates a way to automatically extract ravines using a combination of hydrological analysis and image understanding techniques. A physically realistic hydrological analysis is run on 30-meter DEM data to generate initial estimates of ravine locations, which are then fine-tuned by an active contour (snake) method. The approach builds on standard tools, rather than being implemented as a stand-alone system.

1 Introduction

A key use of simulation and terrain database systems by the military is to conduct training exercises and to plan missions. Currently, however, there is a notable lack of micro-terrain features (features smaller than the scale of the terrain source data) which can have a large impact on the utility of the system. In this paper, we describe initial experimentation with a method to extract ravine micro-features from data comparable to that available in the NIMA data inventory, augmented by georegistered aerial imagery.

Our method is primarily designed to be used on arid geographic regions, where ravines (aka "dry washes" or "wadis") occur due to erosion of the soil during infrequent rainstorms. These ravines typically occur as flat-bottomed corridors with steeply rising sides which range from a fraction of a meter to two to three meters high, though some can be much deeper. As such, they are strategically significant, as they can be used by dismounted infantry as high-speed, concealed avenues for movement, while movement across them can be particularly difficult for mechanized vehicles.

The primary method used in obtaining our goal of extracting ravine micro-features is to use the combination of a hydrological analysis on the DEM (estimating where water flows in this area), combined with a vision-based analysis of the orthophoto (verifying where ravines actually are). Since the washes are mostly cut from erosion due to water, the hydrological analysis hypothesizes likely wash locations. However, the hydrological analysis alone is insufficient to either confirm the actual existence of such features or localize them accurately, because of the restricted depth and small width of the ravines relative to the resolution with which terrain elevation is extracted. Similarly, while ravines are usually at least partially visible on aerial photographs, accurate detection and localization is difficult. The one meter orthophoto contains "too much" information, and ravines are easily confused with roads, tracks and other structures commonly appearing in a desert. However, reliable predictions of exact ravine locations are possible by using the hydrological analysis as an initial estimate, then modifying this estimate using an active contour (snake) method on the orthophoto. The combined approach detects only the features of interest, while simultaneously refining the estimated location. This can be done using single views and does not require redoing complex photogrammetric calculations.

This work was supported by the Defense Advanced Research Projects Agency, contract number pending.

2 Background

The research for this paper is based upon concepts in GIS and hydrological science, and concepts in the area of traditional machine vision.

2.1 Hydrological Analysis for Terrain Analysis

The earliest ideas on using DEM data to find ravines were based upon using local surface properties to look for a part of the topographic surface that is locally concave-upward, and mark this position as a valley or ravine, presuming that it is where surface water runoff is likely to be concentrated (e.g., [Peucker and Douglas, 1975, Chorowicz *et al.*, 1989, Tribe, 1992]).

Many researchers (e.g., [Mark, 1983, Jenson and Domingue, 1988]) have used a method that is more physically justifiable in nature. In this method, a direction is assigned to each cell of the DEM, corresponding to the direction that water would flow out of that cell. This direction is that of steepest decent (i.e. one of the 8 compass directions that corresponds to the steepest downhill slope from that cell). Given this "direction matrix", the total number of cells of the DEM that contribute drainage through each cell is calculated. Those cells that accumulate drainage above a certain threshold are considered part of the drainage network.

2.2 Active Contours for Image Analysis

The original paper on active contours (or snakes) was in 1988 by Kass, Witkin, and Terzopoulos [Kass *et al.*, 1988]. In it, snakes are introduced as energy-minimizing splines, whose energy is a weighted sum of internal and external energies. There are two different internal energies which may be weighted in order to force a snake to act more like a membrane or string, in the sense of it resisting stretching, or more like a thin-plate or rod, in that it resists bending [Leymarie and Levine, 1993]. The external energy equation is a function of the image on which it is acting. This equation can be specified to favor various image properties, such as edges and lines. Snakes bring to bear a high-level, global knowledge across the entire curve, instead of relying solely on local, low-level knowledge[Kass *et al.*, 1988, Menet *et al.*, 1990].

In recent years, much research has been directed at various aspects of snakes, including initialization (e.g., [Berger and Mohr, 1990, Neuenschwander *et al.*, 1994]), different underlying representation of snakes (e.g., [Menet *et al.*, 1990, Wang *et al.*, 1996]), formulation of energy functions (e.g., [Radeva *et al.*, 1995, Lai and Chin, 1993]), imposing constraints (e.g., [Neuenschwander *et al.*, 1995, Fua and Brechbuehler, 1995]),and the method of solution (e.g., [Wang *et al.*, 1996, Amini *et al.*, 1990]).

3 Approach

In our research, we begin with a hydrological analysis of a DEM to form estimates of ravine locations, and then use active contours to attach to the true location of the ravines. We have tested this approach on data of the live fire range (Range 400) area of the USMC Air Ground Combat Center in California, covering a 3.3km by 2.2km area. Using Range 400 as a test area helps with evaluation, since terrain information is available in both standard DTED formats, and in a high-quality DEM with a 1m post spacing and a relative vertical accuracy on the order of 0.1 to 0.3 meters with a matching orthoimage at the same resolution. The results presented here are based on a 30m DEM produced by downsampling the 1m DEM and quantizing elevation values to the nearest meter in order to simulate typical low resolution data. Results based on DTED level 2 data will be presented in subsequent papers.

Figure 1 shows a flowchart depicting the process used to determine ravine locations. Hydrological and IU components provide separate sources of information, which are combined using snakes to produce a final estimate of ravine locations.

3.1 Hydrological Analysis

Hydrological analysis is done using standard tools in the widely available Arc/Info geographic information system (GIS). Localization of drainage patterns is aided by resampling the 30m DEM at a 10m resolution using the Arc/Info `resample` command with cubic convolution (i.e., smoothing over a 4x4 patch). We then use the Arc/Info commands `fill`, `flowdirection`, `flowaccumulation`, `con`, and `streamline` to run a physically realistic hydrological analysis on the 10m interpolated data, in order to obtain our estimates for the location of the

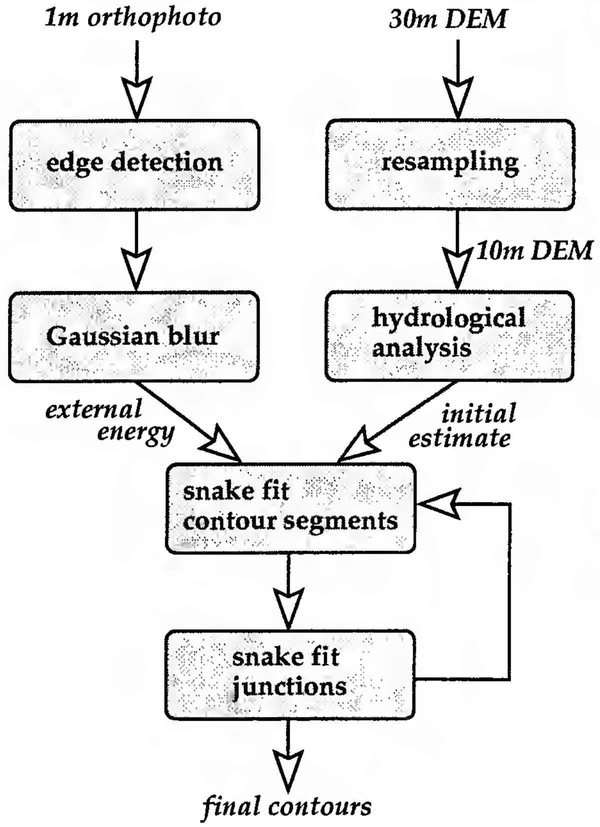


Figure 1: Flowchart ravine location process.

ravines.

3.2 Active Contours

We use polyline (as opposed to B-spline) snakes, solving the energy equations using the discrete dynamic programming method introduced by Amini [Amini *et al.*, 1990]. In this method, local minima are bypassed in favor of the global minimum for the snake, and the snake is guaranteed to converge to a final solution (a position in which it can not be modified and attain a lower energy) within a finite number of iterations [Amini *et al.*, 1990]. The hydrological analysis provides the initial estimate for the polyline snakes. The external energy function is created using a potential function approach based on edge elements computed from the aerial imagery, rather than using image gradients directly. An edge image is produced in which pixels corresponding to all detected edge elements are set to a common value. The edge image is then blurred using a Gaussian kernel with a standard deviation proportional to the expected uncertainty in the initial position estimates from the hydrological analysis. This avoids any need to merge edge elements into longer contours and completely decouples the scale parameter associated with edge detection from the scale parameter specifying uncertainty in the snake initialization.

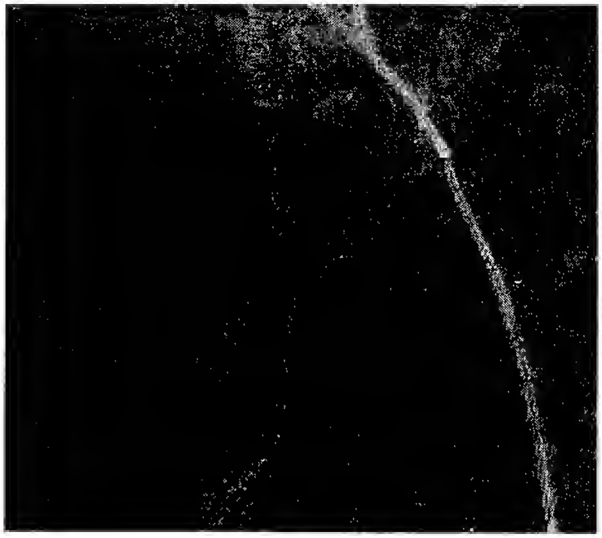


Figure 2: Aerial image of 210m by 240m region of the Range 400 area.

The hydrological analysis produces a tree structured network of flow contours. Snakes are used to refine these contours by alternating between two optimization processes:

- Individual contours are optimized with respect to the external energy function by holding their endpoints fixed.
- Junctions of contours are optimized with respect to the external energy function by simultaneously allowing the nearest 50m of each contour associated with the junction to move.

4 Results

This section describes some preliminary results obtained on data of the Range 400 area. The area that is shown in the figures is a 210m by 240m segment of the Range 400 area.

The image analysis was started using the 1m orthophoto shown in Figure 2. A zero-crossing edge detector was run on this image to produce the edges in Figure 3. These were then blurred using a Gaussian kernel with a standard deviation of 3.0. This result is shown in Figure 4. The 30m DEM

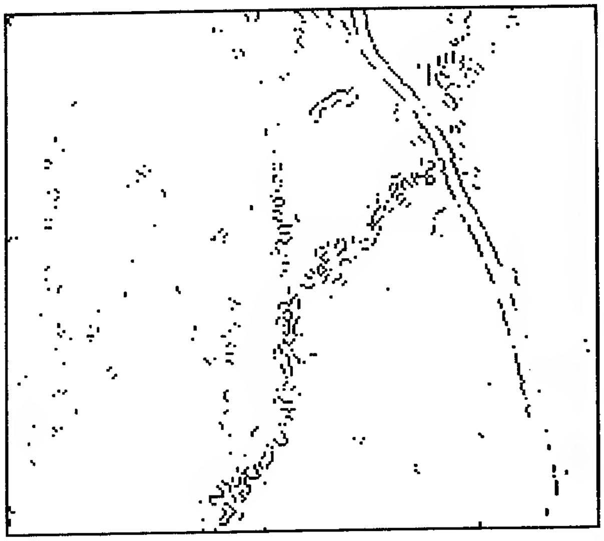


Figure 3: Zero crossing edge detector applied to image in Figure 2.

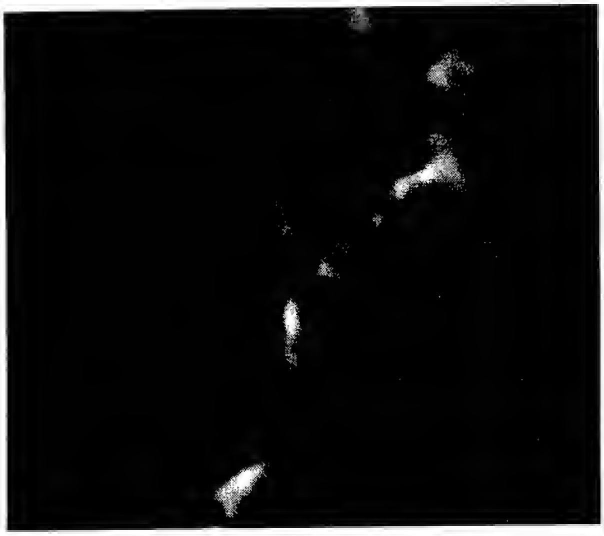


Figure 4: A Gaussian blur of the edge image, with a standard deviation of 3.0.

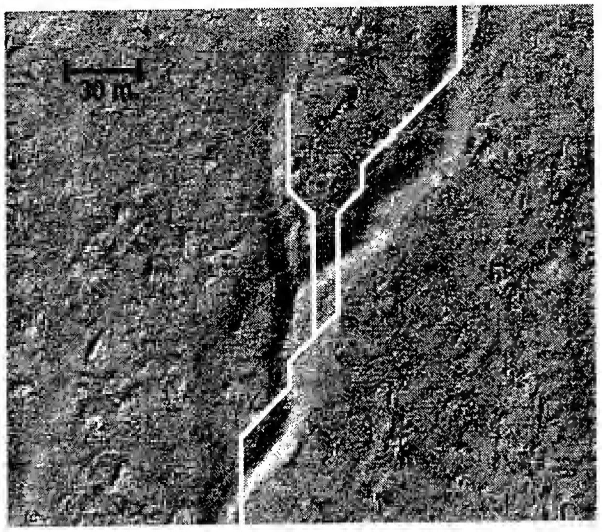


Figure 5: Initial ravine estimates based on hydrological analysis of 10m DEM data.

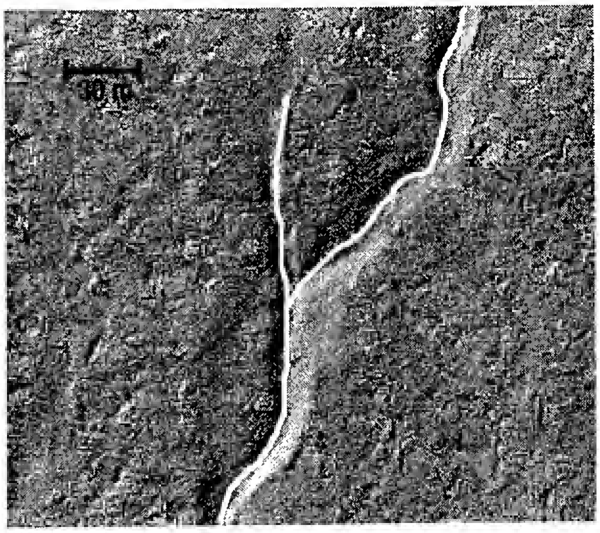


Figure 6: The final contours after fitting with "snakes".

data was interpolated to 10m posts in Arc/Info, then run through the Arc/Info hydrological analysis programs to arrive at the initial contour estimates shown in Figure 5, shown here for purposes of illustration overlayed onto a shaded relief image produced from the 1m DEM data. Figure 6 depicts the final ravine contours after being run through the snaking programs for an average of 148 iterations per contour.

Notice that the road has not been picked out as a ravine, which an image analysis technique by itself would most likely handle incorrectly. Notice also on the orthophoto that there appears to be a number of other objects that appear to be ravines, besides the ravine that we find. Though these appear on the orthophoto and they may even be features produced by the flow of water, they are not present in the high-detail 1m DEM and are thus not of tactical importance. Again, an image analysis technique alone would not be able to distinguish between these easily.

These results on a subsection of the Range 400 area demonstrate the potential for using the combination of hydrological techniques on low resolution DEM data in order to initialize snakes running on high resolution orthophoto data. We believe it provides a useful technique for finding and localizing the location of micro-terrain ravines while avoiding the pitfalls (such as finding spurious contours) of an image analysis technique alone.

References

[Amini *et al.*, 1990] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.

[Berger and Mohr, 1990] M-O. Berger and R. Mohr. Towards autonomy in active contour models. In *10th International Conference on Pattern Recognition*, pages 847–851, Piscataway, NJ, 1990.

[Chorowicz *et al.*, 1989] J. Chorowicz, J. Kim, S. Manoussis, J. P. Rudant, P. Foin, and I. Veillet. A new technique for recognition of geological and geomorphological patterns in digital terrain models. *Remote Sensing of Environment*, 29:229–239, 1989.

[Fua and Brechbuehler, 1995] P. Fua and C. Brechbuehler. Imposing hard constraints on soft snakes. Tech Note 553, Artificial Intelligence Center, SRI International, October 1995.

[Jenson and Domingue, 1988] S. K. Jenson and J. O. Domingue. Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, 54(11):1593–1600, 1988.

[Kass *et al.*, 1988] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[Lai and Chin, 1993] K. Lai and R. Chin. On regularization, formulation and initilization of the active contour models (snakes). In *Second Asian Conference on Computer Vision*, pages 542–545, Singapore, 1993.

[Leymarie and Levine, 1993] F. Leymarie and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):617–634, 1993.

[Mark, 1983] D. M. Mark. Automated detection of drainage networks from digital elevation models. In *Auto-Carto VI: Proceedings Sixth International Symposium on Computer Assisted Cartography*, pages 288–298, 1983.

[Menet *et al.*, 1990] S. Menet, P. Saint-Marc, and G. Medioni. Active contour models: Overview, implementation, and applications. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 194–199, Los Angeles, CA, 1990.

[Neuenschwander *et al.*, 1994] W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler. Initializing snakes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–663, Seattle, June 1994.

[Neuenschwander *et al.*, 1995] W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler. From ziplock snakes to velcro surfaces. In *Automatic Extraction of Man-made Objects from Aerial and Space Images*, pages 105–114. Birkh auser Verlag, 1995.

[Peucker and Douglas, 1975] T. K. Peucker and D. H. Douglas. Detection of surface-specific points by local parallel processing of discrete terrain elevation data. *Computer Graphics and Image Processing*, 4:375–387, 1975.

[Radeva *et al.*, 1995] P. Radeva, J. Serrat, and E. Marti. Snakes for model-based segmentation. In *Fifth International Conference on Computer Vision*, pages 816–821, Cambridge MA, 1995.

[Tribe, 1992] A. Tribe. Automated recognition of valley lines and drainage networks from grid digital elevation models: A review and a new method. *Journal of Hydrology*, 139:263–293, 1992.

[Wang *et al.*, 1996] M. Wang, J. Evans, L. Hassebrook, and C. Knapp. A multistage, optimal active contour model. *IEEE Transactions on Image Processing*, 5(11):1586–1591, 1996.

Quantitative Comparison of IU Algorithms

Eric S. Jensen and **William B. Thompson**
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
http://www.cs.utah.edu/projects/robot/benchmark.html

Abstract

This paper presents a procedure for quantitatively comparing the performance of two image understanding algorithms using real images. We compare each algorithm's reconstructed points with the "ground truth" object points acquired with a laser 3–D position digitizer. This procedure is necessary because qualitative comparisons are not precise enough to identify small errors and synthetic images do not properly test algorithms that are intended for use on real imagery. We outline the procedure we use for comparing stereo reconstruction algorithms and show an example.

1 Introduction

Much discussion has been given over to the need for ways to evaluate image understanding algorithms and systems in an objective and quantitative manner [Sawhney and Hanson, 1990, Weems *et al.*, 1991, Firschein *et al.*, 1993, Bolles *et al.*, 1993]. A few calibrated datasets are now available (e.g., see [Willson and Shafer, 1992, Thompson and Owen, 1994, Owen *et al.*, 1996]), but reports of either quantitative evaluations of IU methods or the procedures for performing such evaluations on real imagery are still rare.

Informal, subjective measures of performance can suffice if the manifestation of errors is large. If the results "look wrong," they probably are. Evaluation is *much* harder when errors are more subtle or when two methods with similar performance need to be compared. Often, synthetic imagery is used in such situations (e.g., [Barron *et al.*, 1994]). However, there are well known problems with using synthetic imagery to evaluate IU methods that are intended for use with real images. This paper presents a procedure for determining the relative performance of two vision algorithms. Our aim is neither to draw broad conclusions about the specific algorithms considered nor to deal in any depth with the mathematical issues involved in quantifying error. Rather, we simply want to show the engineering steps necessary to provide the framework for such activities.

Evaluating image understanding methods that determine scene geometry requires that the "true" geometry be independently known. Further, there must be an effective way to compare this representation of geometry with that produced by the algorithm under investigation. Determining the true scene geometry involves either using objects of known shape and accurately measuring the position, or scanning the scene with some sensing device presumed to be significantly more accurate than the vision system being tested [Owen *et al.*, 1996]. In the procedure presented here, a laser scanner is used to measure scene geometry in order to evaluate how accurately two stereo reconstruction algorithms are able to estimate depth.

2 Gathering Data

A set of test data consists of the image pairs needed by the stereo algorithms and the ground truth to compare against. Results from stereo algorithms are often represented as a camera-centered *depth map*. In order to directly compare the stereo results with

This work was supported by the Advanced Research Projects agency under Army Research Office grant number DAAH04-93-G-0420.

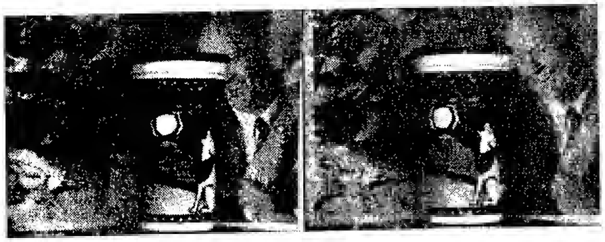


Figure 1: Stereo image pair.

the true geometry, the ground truth should also be rendered into a camera-centered depth map. Test data could potentially be acquired using two cameras capable of sensing both intensity and range, but such devices produce imagery that differs in both geometry and photometry from standard cameras. We chose to use conventional cameras and camera calibration techniques, augmented with the use of a DIGIBOT II laser 3–D position digitizer. The DIGIBOT takes precise measurements of 3–D position points on the surface of objects. The objects sit on a platter which the DIGIBOT can rotate in order to scan the objects from all angles. Such a capability is essential if ground truth depth maps are to be produced for each camera viewpoint.

The first step in the process of gathering test data requires a calibration target to be placed within the workspace of the DIGIBOT so it is visible from both cameras. The calibration target we use is a dark plastic cube 75 mm on a side, with 24 white dots inset around the outside of each face, centered 7.5 mm from the edge and 10 mm from each other. An image of the cube is captured from each camera, and 3–D position points on the surface of the cube are scanned by the DIGIBOT. The dots in the images can easily be located to sub-pixel accuracy. From the location of the dots and the known geometry of the cube, we calculate the camera models in a coordinate system attached to the calibration cube using standard methods for optically calibrating cameras [Tsai, 1987]. Using the DIGIBOT data, the position of the planar faces of the cube in the DIGIBOT coordinate system can be determined with high precision. Combining these results, we can transform between DIGIBOT coordinates and camera coordinates.

The second step in the process involves replacing the calibration target with some configuration of objects of interest. As before, stereo images are captured, and the scene objects are scanned by the DIGIBOT. The scanned 3–D points are connected into a surface representation using a triangulated mesh specified in terms of the DIGIBOT coordinate system. This mesh is transformed into the camera coordinate system. It can then be projected into each camera's image plane using standard rendering techniques, except that in this case it is the z values that are rendered and not intensity. The result is a per-pixel depth map for each camera.

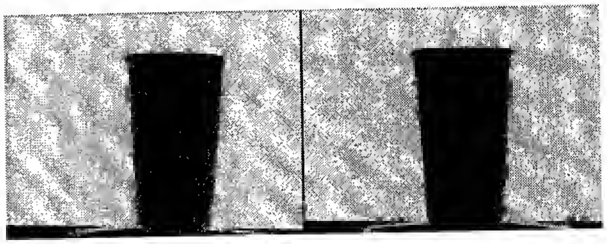


Figure 2: "Ground truth" range images for left and right camera views.

3 Comparison of Stereo Algorithms

As an example, we compared two stereo algorithms. Algorithm A used the method described in [Smitley and Bajcsy, 1984]. Algorithm B is similar, but it skipped some steps to make it run faster. Using the procedure described above to gather test data, we compared the algorithms to see how much quality is lost when using the faster algorithm. The stereo images used are shown in Figure 1. Each stereo algorithm was used to reconstruct 3–D point data from the images. Though we have not done so, repeated experiments could be used to determine the statistical significance of the differences.

Left Image	Algorithm A	Algorithm B
bad matches	804	1372
good matches	10138	10196
ave. error (mm)	1.496	1.510

Right image	Algorithm A	Algorithm B
bad matches	844	1051
good matches	10093	9779
ave. error (mm)	1.487	1.494

Table 1: Performance of two stereo algorithms. Good matches have less than 5 mm error.

Range values from the two stereo reconstructions were compared, on a per-pixel basis, to the range images constructed from the DIGIBOT scan (see Figure 2), but only at those pixels for which the stereo algorithm was able to determine a match. Reconstructed depth values were classified as a *good match* if they were within 5 mm of the true value and a *bad match* otherwise. The quantitative RMS error associated with the good matches and the number of good and bad matches are reported in Table 1. The distribution of good and bad matches is shown in Figure 3.

Algorithm A is clearly better then Algorithm B (see Table 1, Figure 3, and Figure 4). Not only did Algorithm A have fewer bad matches, it also had more good matches and a slightly lower average error for the good matches.

When working to improve IU algorithms it is useful to be able to determine when one version of an algorithm performs even slightly better than another version on real images. The procedure described here is a fairly simple way to do such comparisons.

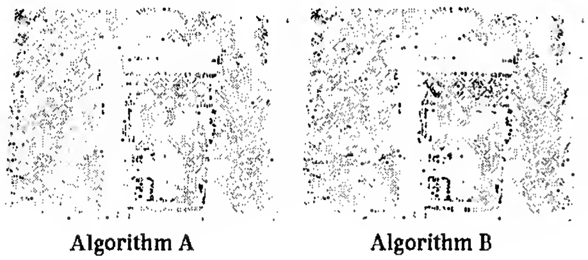


Figure 3: The location of good matches (gray) and bad matches (black).

References

[Barron *et al.*, 1994] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, pages 43–77, February 1994.

[Bolles *et al.*, 1993] R. C. Bolles, H. H. Baker, and M. J. Hannah. The JISCT stereo evaluation. In *Proc. ARPA Image Understanding Workshop*, April 1993.

[Firschein *et al.*, 1993] O. Firschein, M. A. Fischler, and T. Kanade. Creating benchmarking problems in machine vision: Scientific challenge problems. In *Proc. ARPA Image Understanding Workshop*, April 1993.

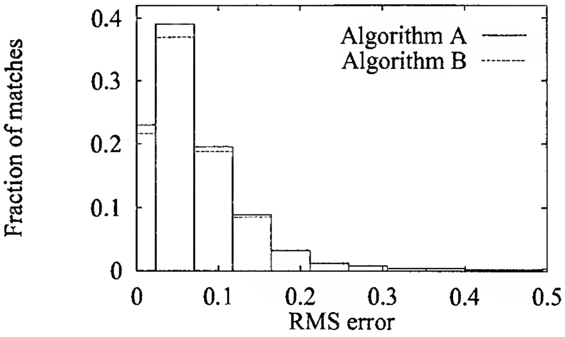


Figure 4: Histogram of errors in the left image.

[Owen *et al.*, 1996] J. C. Owen, H. J. de St. Germain, S. Stark, T.C. Henderson, and W. B. Thompson. Calibrated imagery for quantitative evaluation of IU pose-estimation and stereo algorithms. In *Proc. ARPA Image Understanding Workshop*, pages 987–994, February 1996.

[Sawhney and Hanson, 1990] H. S. Sawhney and A. R. Hanson. Comparative results of some motion algorithms on real image sequences. In *Proc. DARPA Image Understanding Workshop*, September 1990.

[Smitley and Bajcsy, 1984] D. L. Smitley and R. Bajcsy. Stereo processing of aerial, urban images. In *Proc. Seventh Int. Conference on Pattern Recognition*, pages 433–435, 1984.

[Thompson and Owen, 1994] W. B. Thompson and J. C. Owen. "Hard-copy" benchmark suite for image understanding in manufacturing. In *Proc. ARPA Image Understanding Workshop*, pages 221–227, November 1994.

[Tsai, 1987] R. Y. Tsai. A versitile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

[Weems *et al.*, 1991] C. C. Weems, A. R. Hanson, E. M Riseman, and A. Rosenfeld. The DARPA image understanding benchmark for parallel computers. *Journal of Parallel and Distributed Computing*, 11(1):1–24, January 1991.

[Willson and Shafer, 1992] R. G. Willson and S. A. Shafer. Precision imaging and control for machine vision research at Carnegie Mellon University. In *Proc. SPIE Conf. on High Resolution Sensors and Hybrid Systems*, February 1992.

SECTION V
AUTOMATIC TARGET RECOGNITION
(ATR)

Automatic Target Recognition (ATR) Principal Investigator Reports

Model-Based Target Recognition in Foliage-Penetrating SAR Images

R. Chellappa† P. Burlina† J. Song‡
†Center for Automation Research, University of Maryland
College Park, MD 20742-3275 (rama@cfar.umd.edu)
‡DEMACO, Inc., Champaign, IL 61820

Abstract

This paper describes plans for developing a model-based system for target recognition in foliage. This approach is based on using the phenomenology of the Foliage Penetrating (FOPEN) sensor, and on resonance-based features for recognition. This paper describes the key components of our FOPEN SAR ATR system, the research problems that will be addressed, and performance evaluation plans.

1 Introduction

We describe a joint research effort by the University of Maryland (UMD) and DEMACO, aimed at developing a model-based ATR system for recognition of targets in FOPEN SAR images. Detection and recognition of targets in FOPEN SAR is a prime example of ATR in very low signal/clutter ratio situations. Since returns from clutter can be as strong as those from targets, and the clutter is non-Gaussian, the detection problem poses serious challenges. Also, in this frequency region, the incident wavelength is of the same order as the size of the body, leading to a resonant behavior of the targets. Attempts to recognize targets in foliage have used one or another form of matching to templates derived from target resonances [3,10,11,12,14,15]. Most of the recent target-resonance-based approaches have been bottom-up approaches that made no use of explicit models or model-derived signatures. In programs such as MSTAR, SAIP and RADIUS, explicit use of 3-D models of sites and targets as well as sensor phenomenology have played a key role in providing context and model-based signatures that can be subsequently used for focus of attention, clutter mitigation and matching. Such model-based approaches are sorely needed for the FOPEN SAR ATR problem, as the phenomenology and very low signal/clutter ratio make the problem even more difficult when compared to the high-frequency SAR case.

The main objective of this project is to study a model-based approach to recognition of targets in foliage. Specifically, we will investigate target-resonance-based approaches to recognition. A key component in this approach is the creation of high-quality synthetic FOPEN images from which templates are created. Due to the non-availability of simulation algorithms for FOPEN signatures that produce acceptable quality [14,15], templates generated from exemplars are often used. DEMACO will develop new techniques for synthetic generation of radar images at low frequencies (less than 1 GHz) and demonstrate the utility of predicted signatures in detecting and recognizing targets in foliage.

Our model-based FOPEN SAR ATR framework consists of four key components: Focus of Attention (FOA), FOPEN SAR target and foliage signature prediction, feature extraction, and recognition. The FOA module includes algorithms for segmentation, labeling and detection. The segmentation and labeling component will divide the FOPEN SAR images into clear, medium and dense foliage regions. The detection algorithm will adaptively threshold the early-time response depending on whether the target is in a clear area or in foliage. The prediction module will generate synthetic images of targets buried in foliage. The matching module will use the predicted signatures for matching image-derived resonance-based features to model-derived features. Resonance is one of the peculiar features of the FOPEN imaging modality, whereby an early optical time response is always followed down-range

This project will be supported by the Defense Advanced Research Projects Agency (ARPA Order No. E656) and the U.S. Army Research Laboratory. For further information see
http://www.cfar.umd.edu/~akunuri/FOPEN/fopen.html.

by a late-time resonant response which depends essentially on the target dimensions.

One of the much-studied approaches to the recognition of targets in FOPEN images is the method of resonance extraction known as the singularity expansion method (SEM) [3], a contemporary variation of Prony's method. Since SEM results in poor performance in the presence of noise and multipath effects, spectral correlation methods have been proposed [14,15]. In this approach, synthetic and image-derived resonances are transformed to a spectral domain (Fourier, DCT or wavelet basis) and correlated. This approach has been tested on canonical targets as well as images containing CUCVs and other confuser targets. Both targets in the open and targets in foliage have been considered.

One of the key issues to be solved in this approach is the generation of synthetic reference signals. As algorithms for predicting resonances for complicated targets are not yet available, reference signatures have been generated using pole patterns through the resonant portions of the CUCV signatures in two different images.

Another aspect of resonance-based ATR for FOPEN is the claim that target resonances are invariant to aspect. This is only approximately true. This is because the locations of the poles in frequency space are fully invariant, but the residues are a function of illumination, which in turn is a function of aspect angle relative to the radar. If several targets are in the foliage, then matching using only one template/target may not be adequate due to clutter corruption and signal attenuation. New matching techniques for the multi-target problem need to be developed.

Our planned system will incorporate the following components:

Prediction of low-frequency SAR signatures. Until now Xpatch has had a lower frequency limit below which it was not effective. The existing ray tracing technique in Xpatch will be improved to handle foliage scattering. More importantly, DEMACO will develop novel techniques for synthetic generation of target and foliage signatures at low frequencies, using the exact Method of Moments (MoM) technique. This component alone will provide new approaches to model-based recognition of targets in foliage.

Ongoing SAR ATR efforts at UMD heavily utilize Xpatch-generated signatures. Xpatch is a radar signature prediction code developed by DEMACO and is based on a high-frequency asymptotic method called "shooting and bouncing rays". For computing SAR images of military targets, Xpatch has been validated for the frequency range of S-band and above. For lower frequencies, however, we do not expect Xpatch to give good results unless it is supplemented by new techniques suitable for describing low-frequency resonant phenomena. Prediction of SAR images of targets and the surrounding clutter at 1 GHz and below requires both high- and low-frequency codes. We wish to compute SAR images of ground vehicles in a clutter environment including foliage canopies. In terms of EM wavelength, the problem has two scales: large-scale in clutter and small-scale in targets. For penetration through canopies, the existing ray-tracing approach is sound as long as proper physics associated with 1 GHz wave propagation is added to Xpatch. The net effect of the canopy is to distort and attenuate the incoming (exiting) rays from the radar. Once rays reach the ground target, a different phenomenon takes over. The typical size of a ground target is 10 meters or 30 wavelengths at 1 GHz. Components on the target that account for the return in a SAR resolution cell are on the order of one wavelength. Hence scattering by target components falls into the so-called "resonance region", which cannot be adequately described by the high-frequency techniques used in Xpatch. Thus, we propose to attack the low-frequency target scattering problem by using a new exact solver code called Fast Illinois Solver Code (FISC), which is based on the state-of the-art fast multipole method. Similarly to Xpatch, FISC computes radar signatures and images from a complex target represented by a realistic CAD geometry file. But the method used in FISC is the exact MoM instead of the asymptotic ray tracing technique.

Model-based false alarm reduction. Despite our best efforts to optimally set detection thresholds, a large number of false alarms occur due to the very low signal/clutter ratio. Current attempts at false alarm reduction use empirical target images obtained from other collections. We intend to use predicted signatures to derive target attributes that can be exploited to eliminate non-target clusters which often yield false alarms.

Model-based matching. We plan to exploit the predicted signatures to derive templates for matching target and reference features, and to explore spectral correlation matching and other techniques based on pole profiles. We will do a systematic study of how quasi-aspect-invariance of target resonances will impact matching complexity.

Our long-term goal is to study a model-based approach to the FOPEN SAR ATR problem. Fig. 1 shows the various modules including Focus of Attention, feature extraction, prediction, indexing, matching, and search [5]. The FOA module could propose target chips to the feature extraction mod-

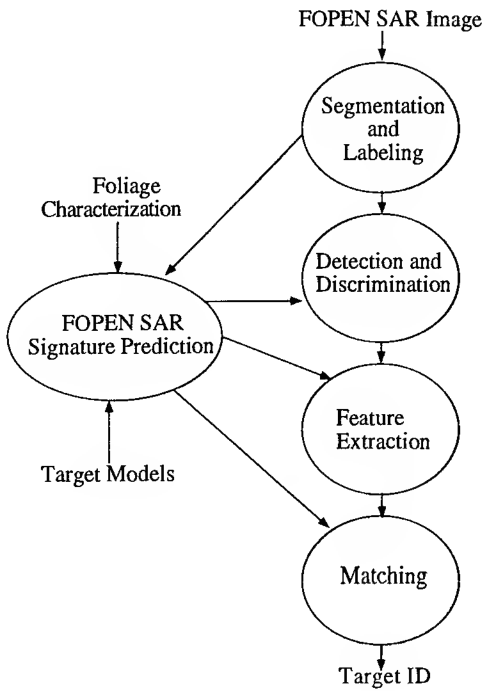


Figure 1: FOPEN SAR ATR system architecture

ule. Since low-frequency EM phenomenology is very different, issues related to feature extraction are still unresolved. Likewise, the prediction and matching modules, although they will serve the same roles as their counterparts in the MSTAR program, will be significantly different owing to the low frequencies employed and the foliage that must be taken into account. Also different types of invariance come into play for features based on target resonances.

Our FOA module will be based on our research efforts with the Army Research Laboratory in the context of the Federated Laboratory Advanced Sensor Consortium. The next section of this paper describes the various system components and open research issues, and discusses our performance evaluation plans.

2 Open Research Issues and Critical System Components

UMD and DEMACO propose to undertake a systematic study of IU approaches to FOPEN SAR. Our work is based on four research problems: Focus of attention for FOPEN SAR images, prediction of model-based target signatures at FOPEN frequencies, model-based false alarm reduction, and matching.

2.1 Prediction of low-frequency SAR signatures

DEMACO will undertake tasks involving the prediction of target resonances using MoM. By using MoM, the EM problem is reduced to solving an N-by-N matrix equation, where N is the number of unknown current samples induced on the target. For an airplane, N is typically 10,000 at 0.1 GHz, and it increases with the frequency squared, which leads to two important conclusions: (a) Within current computer resources, it is very difficult to use MoM for military targets beyond 0.1 GHz for SAR image simulations. (b) An efficient MoM-based code must have a good algorithm for solving the matrix equations rapidly for many frequencies.

The standard algorithm for solving matrix equations is LU Decomposition (LUD). LUD is an $O(N^3)$ algorithm. In 1990, Rokhlin [13] introduced FMM for solving matrix equations derived from MoM (see also [6]). This was considered a major breakthrough in low-frequency techniques. The FMM idea is to first divide the subscatterers into groups. Then, the addition theorem is used to translate the scattered fields of different scattering centers within a group into a single center (aggregation). Hence, the number of scattering centers is reduced. Similarly, for each group, the field scattered by all the other group centers can be first "received" by the group center, and then "redistributed" to the subscatterers belonging to the group (disaggregation).

It has been proved that the computational cost of two-level FMM is of order $O(N^{1.5})$ [6]. Numerical simulations also show that the complexity is of order $O(N^{1.5})$ [16,17]. A further improvement in FMM is the Multilevel Fast Multipole Algorithm (MLFMA) developed at the University of Illinois. To implement MLFMA, the entire object is first enclosed in a large cube, which is partitioned into eight smaller cubes. Each subcube is then recursively subdivided into smaller cubes until the edge length of the finest cube is about half a wavelength. Cubes at all levels are indexed. At the finest level, we find the cube in which each basis function resides by comparing the coordinates of the center of the basis function with those of the center of the cube. We further find nonempty cubes by sorting. Only nonempty cubes are recorded using tree-structured data at all levels [2,9]. Thus, the computational cost depends only on the nonempty cubes.

The basic algorithm for matrix-vector multiplication is broken down into two sweeps [1]. The first sweep consists of constructing outer multipole expansions for each nonempty cube at all levels. The second sweep consists of constructing local multipole expansions contributed from the well-separated cubes at all levels. When the cube becomes larger as one progresses from the finest level to the coarsest level, the number of multipole expansions should increase. In the first sweep, the outer multipole expansions are computed at the finest level, and then the expansions for larger cubes are obtained using shifting and interpolation.

At the coarsest level, the local multipole expansions contributed from well-separated cubes are then calculated in the second sweep. The local expansions for smaller cubes include the contributions from the parent cube and from cubes that are well-separated at this level but not well-separated at the parent level [4]. At the finest level, the contributions from non-well-separated cubes are calculated directly. Since only nonempty cubes are considered, the complexity of MLFMA is reduced to $O(N\log N)$, and the memory requirements for MLFMA are of order $O(N\log N)$. Dembart and Yip [7,8] have implemented MLFMA using radiation functions, interpolation and filtering; this implementation has complexity $O(N \log^2 N)$.

For large N, the advantage of MLFMA over LUD is very dramatic. Let us illustrate this by an example. Consider an airplane whose dimensions are $15\times9\times4$ meters. At 0.1 GHz, a matrix equation with N=25,000 is set up by MoM, from which we compute its RCS for 182 different incidence angles. Results calculated by FISC using one processor on an SGI Challenge (25 MFLOPS) and results measured by EMCC (the ElectroMagnetic Code Consortium) are in excellent agreement. The memory requirements and speed of FISC and LUD are compared in the following table:

	Speed (Hrs)	Memory(MB)
FISC	45	167
LUD	400	5,200
Savings factor	9	31

We cannot find a computer with 5.2 GB of memory without going to a very large supercomputer, so the above data for LUD are only an estimate. This example illustrates the big benefit of FISC, namely, it makes MoM practical for airplane and tank scattering problems at frequencies below 1 GHz using workstation-sized machines.

DEMACO will develop methodologies for calculating SAR images of realistic ground targets in clutter environments at 1 GHz and below, with special emphasis on foliage penetration. The approach is to use: (a) High-frequency ray tracing techniques in Xpatch to determine attenuation and phase distortion of incoming (exiting) radar waves due to clutter, and (b) FISC, low-frequency MoM, to calculate target returns.

Specifically, we will consider the following problems:

Hybridize low- and high-frequency contributions. Let us denote the scattering contribution from clutter by A and that from the target by B. The total contribution is not simply A+B. How to combine them without double-counting or missing interactions is a challenging research problem. This problem will be addressed in this task.

Speed up frequency looping in FISC. Unlike high-frequency codes, all low-frequency codes are inefficient in repeated frequency calculations. For FISC, a change of frequency, however slight, is a new run. Thus, to form a SAR image with 32 frequency points, the computational cost of FISC is 32 times that of a single-frequency run. This deficiency must be improved before FISC can be used for SAR applications.

Improve foliage scattering in Xpatch. The existing ray-tracing technique in Xpatch models each leaf and branch of a tree individually. This is not feasible for a large cluster of trees or for low frequencies. A statistical model-based on random medium theory and measurements will be researched and implemented in Xpatch to support this development. This overall research and development effort will be a major advance in expanding current computational capabilities across a very wide band of frequencies. Until now Xpatch has had a low frequency limit below which it was not effective. With the advent and further development of FISC, coupled with Xpatch, the new computational technology will provide a wider frequency range for FOPEN ATR research and development.

2.2 Matching and recognition

Several challenging problems remain to be solved. As mentioned earlier, target resonances form the basic attributes that will be used in recognition. Poles extracted from target resonances have been historically used [3,10,11,12] as features for identification. More recently [14,15], using a projection of target resonances onto a transform basis set, a set of coefficients known as spectral templates has been derived. The FOPEN image data are also projected onto the same basis, and the resulting spectral template is correlated with the target template. We propose to evaluate the pole and the transform projection techniques mentioned above using the model-derived signatures. In addition to avoiding the necessity of deriving reference templates from other empirical collections, the ability to use model-derived templates allows us to rapidly test new target and foliage models. Unlike the situation in FLIR and SAR ATR problems where obscuration and concealment are afterthoughts, in the problem under considera-

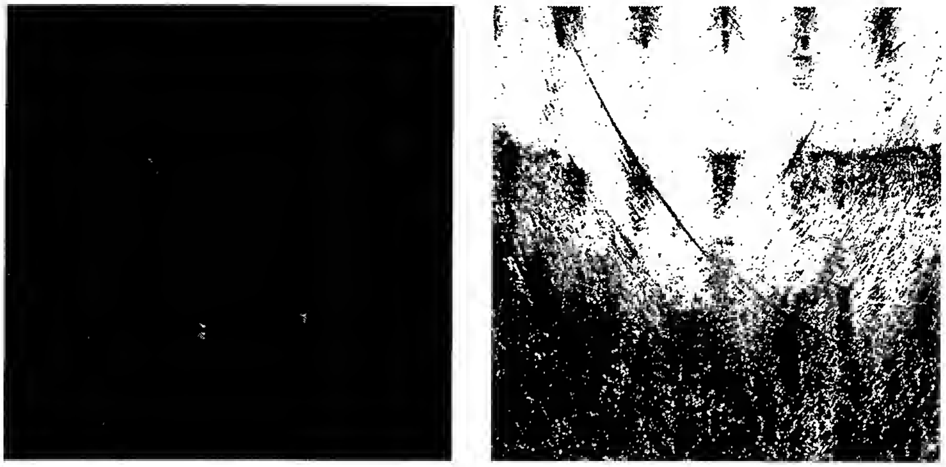


Figure 2: Segmentation of a FOPEN image into clear and foliage regions

tion, targets are buried in foliage. The availability of model-based prediction tools as envisioned in our work enables us to account for the presence of foliage. For example, it has been observed [15] that in the presence of foliage clutter, in addition to substantial clutter corruption, the target signature is attenuated by several dB. Our matching techniques will account for such signature changes.

One of the interesting features of target resonances is that they are quasi-aspect-independent. In the presence of clutter since different amounts of corruption and attenuation may occur depending on the nature of the clutter and the targets, more than one template per target may be required. We will systematically study this issue for single and multiple target cases using data collected by ARL. As pointed out in [18], several questions remain to be answered in FOPEN ATR design. These concern the distinctiveness of pole patterns or spectral templates among the targets under consideration, the stability of resonances during operation, and the sensitivity of polarization and multipath effects.

An example of segmentation into clear and foliage areas used in our FOA scheme is shown in Fig. 2. Fig. 3 shows the positions of detected and classified targets using a spectral matching technique.

2.3 Performance evaluation

The evaluation metrics to be used for FOPEN SAR ATR are the probability of recognition and false alarm rate per square km for the detection modules. For recognition we will produce confusion matrices. The data collected by ARL researchers has ground truth information on target types, locations, articulation, and foliage, which will be useful for evaluating the detection and recognition algorithms.

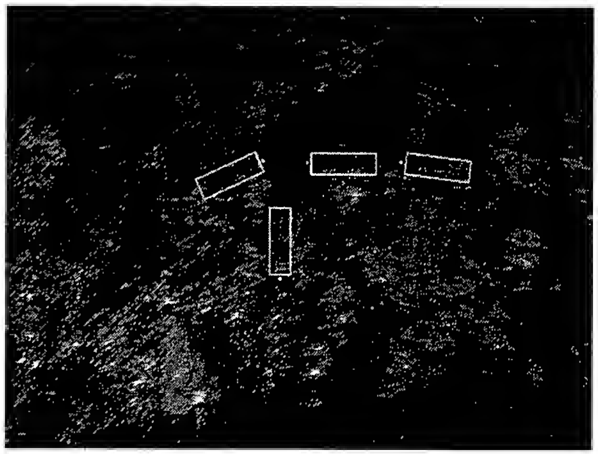


Figure 3: An example of target detection in FOPEN images [14,15]

The evaluation of MoM and other CEM codes has been relatively straightforward because the DoD/NASA Electromagnetic Code Consortium (EMCC) has developed and continues to develop better and better test cases. Using stereo lithography, an accurate CAD model for a VFY218 scale model aircraft has been built. The EMCC has measured this aircraft in an anechoic chamber from 2 to 35 GHz, providing for an equivalent frequency range of 70 MHz to approximately 1 GHz. This wideband data is very useful in validating the FISC code. In addition, this geometry is a benchmark for EM codes, and it will be possible to compare this code's results to other codes when they are capable of running a 1 GHz equivalent test. Currently

FISC is the only code available that can run this problem on a workstation in reasonable time. In addition, Wright Laboratory and others have extensive chamber-measured as well as full-scale vehicle data on the M-35 truck; we will utilize this data for the full evaluation of the code for ground vehicles. Also present in this data set is the M-35 in varying foliage, both in the anechoic chamber as well as at full scale.

3 Conclusion

This paper describes our research plans whose ultimate goal is the design of a system for the exploitation of FOPEN SAR imagery. Our system will emphasize model-based detection coupled with effective target signature prediction.

Bibliography

1. C. R. Anderson, "An Implementation of the Fast Multipole Method Without Multipole," SIAM J. Sci. Stat. Comput., vol. 13, pp. 923-947, July 1992.

2. J. Barnes and P. Hut, "A Hierarchical $O(N\log N)$ Force-Calculated Algorithm," Nature, vol. 324, pp. 446-449, Dec. 4, 1986.

3. C. E. Baum, "The Singularity Expansion Method", in *Transient Electro-Magnetic Fields*, L.B. Felsen (ed.), Chapter 3, Springer, New York, 1976.

4. A. Brandt, "Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernels," Comput. Phys. Commun., vol. 65, pp. 24-38, 1991.

5. R. Chellappa et al., "Model-Supported Exploitation of SAR Images," Proc. DARPA Image Understanding Workshop, Palm Springs, CA, pp. 389-407, Feb. 1996.

6. R. Coifman, V. Rokhlin, and S. Wandzura, "The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription," IEEE Antennas and Propagation Magazine, vol. 35, no. 3, pp. 7-12, June 1993.

7. B. Dembart and E. Yip, "A 3D Moment Method Code Based on Fast Multipole," Digest of the 1994 URSI Radio Science Meeting, p. 23, Seattle, WA, June 1994.

8. B. Dembart and E. Yip, "A 3D Fast Multipole Method for Electromagnetics with Multiple Levels," 11th Annual Review of Progress in Applied Computational Electromagnetics, vol. 1, pp. 621-628, Monterey, CA, March 1995.

9. L. Hernquist, "Performance Characteristics of Tree Codes," Astrophysical J. Supp., vol. 64, pp. 715-734, 1987.

10. R.H. Mains and D.L. Moffatt, "Complex Natural Resonances of an Object in Detection and Discrimination" TR-3424-1, Electroscience Laboratory, Ohio State University, Columbus, OH, June 1974.

11. M.A. Morgan and P.D. Larison, "Natural Resonance Extraction from Ultra Wide Band Scattering Signatures", in *Ultra Wideband Radar: Proc. of the First Los Alamos Symposium*, B. Noel (ed.), pp. 203-215, CRC Press, Boca Raton, FL, 1991.

12. L.W. Pearson, M.L. VanBlaricum, and T. Mittra, "A New Method for Radar Target Recognition Based on the Singularity Expansion for the Target", Proc. IEEE International Radar Conference Record, pp. 452-457, April 1975.

13. V. Rokhlin, "Rapid Solution of Integral Equations of Scattering Theory in Two Dimensions", J. Computational Physics, vol. 86, pp. 414-439, Feb. 1990.

14. V. Sabio, "Spectral Correlation of Wideband Target Resonances", Algorithms for Synthetic Aperture Radar Imagery 3, SPIE Proc. vol. 2757, pp. 145-151, April 1996.

15. V. Sabio and R. Chellappa, "Efficient Method of Target Recognition Based on Spectral Correlation of Wideband Resonance Effects," Algorithms for Synthetic Aperture Radar Imagery, SPIE Proc. vol. 2230, pp. 328-335, April 1994.

16. J.M. Song and W.C. Chew, "Fast Multipole Method Solution Using Parametric Geometry," Micro. Opt. Tech. Lett., vol. 7, pp. 760-765, Nov. 1994.

17. J.M. Song and W.C. Chew, "Fast Multipole Method Solution of Combined Field Integral Equation," 11th Annual Review of Progress in Applied Computational Electromagnetics, vol. 1, pp. 629-636, Monterey, CA, March 1995.

18. M.L. VanBlaricum, "Radar Cross Section and Target Scattering", in *Introduction to Ultra-Wideband Systems*, J.D. Taylor (ed.), pp. 457-490, CRC Press, Boca Raton, FL, 1995.

Feature Extraction using Attributed Scattering Center Models for Model-Based Automatic Target Recognition

Randy Moses and Lee Potter*
Department of Electrical Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210
E-MAIL: {randy,potter}@ee.eng.ohio-state.edu
HOMEPAGE: http://er4www.eng.ohio-state.edu/ips/research/Projects/darpa/dapra.html

Abstract

This report summarizes our research plans and direction for the above-named DARPA IU program. The program begins in March 1997. We discuss the research objectives, the research questions to be considered, and our performance evaluation process.

1 Objectives

Our program is aimed at developing improved feature extraction for application in feature-based automatic target recognition (ATR) systems. Our focus is on synthetic aperture radar (SAR) ATR, although our methods also apply to non-SAR radar ATR, such as high range resolution radar ATR.

Our primary research objectives are:

1. **Develop and validate physically-based models for scattering that can be used for model-based ATR.** Our work is aimed at developing attributed scattering center models for radar scattering. These models are founded on electromagnetic scattering theory (uniform theory of diffraction and physical optics). We base our models on scattering theory to ensure that the derived models are physically meaningful.

   Our approach is to distill from these electromagnetic scattering theories models that balance between fidelity and simplicity. The electromagnetic theories contain much more detail of scattering behavior than is practical for ATR applications. Our goal is to obtain simplified approximations that contain few enough parameters to be reliably estimated from measured data. To do so, we build on our past successes as described, for example, in [Potter and Moses, 1997].

2. **Develop practical feature estimation algorithms.** The aim is to derive estimation algorithms that can be implemented in SAR ATR systems. Such algorithms should be computationally practical, should determine model order autonomously, and should be "hands off" algorithms, requiring little or no hand-tuning by the user.

   Our effort will be focused on developing algorithms that estimate model parameters from complex-valued SAR imagery in the image domain. This is in contrast to many existing parameter estimation methods that operate on SAR phase history data. Image domain processing is more practical, because existing ATR systems operate on SAR image "chips" that have been prescreened by front-end processing.

*This work was sponsored by the Defense Advanced Research Projects Agency under contract F33615-97-1020 monitored by Wright Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

In addition, we will consider model order detection and image segmentation algorithms needed for feature extraction. Finally, computational speed of the algorithms is of importance; we will aim to develop algorithms with good statistical properties while minimizing computation.

3. **Measure efficacy of these features for improvements to SAR ATR performance.** The goal is to quantify the ATR performance improvement afforded by our new models and estimation algorithms. We will quantify improvement first by determining the uncertainty of the estimated features. Feature uncertainty is needed in Bayesean evidence accrual used for scoring of candidate target hypotheses to the measured target data. We will consider algorithm-independent bounds, such as the Cramér-Rao bound, as well as algorithm-dependent performance. In addition, we will quantify ATR detection performance performance using the feature estimates in conjunction with feature-based match scoring.

The research program builds on past ATR work funded by a DARPA University Research Initiative. Our current technical approach is based on this past work, as summarized in [Potter and Moses, 1997, Ying, 1996, Chiang, 1996].

Our program is of value to battlefield awareness because we improve SAR ATR performance in the following ways:

1. **Richer feature set for ATR:** Our physically-relevant models characterize scattering using multiple attributes for each scattering center; this gives an increased level of target discriminability, which improves ATR performance. Richer features are especially important in extended operating conditions, when fewer target scattering centers are observable.

2. **Decreased feature uncertainty:** Because our models are physically based, we are able to exploit prior information in the feature extraction stage to improve resolution. We can thus achieve sub-pixel accuracy and superresolution of target scattering phenomena. The smaller the feature uncertainty, the more discriminable are targets, and ATR performance improves.

3. **Improved match scoring metrics:** We will develop match scoring rules that are tailored to the feature extraction algorithms we develop. By characterizing feature uncertainty and coding this uncertainty in a match score, we will further increase target discriminability.

The above performance improvements have application in feature-based ATR systems, such as MSTAR. We are closely following the MSTAR program, and are tailoring the research program to facilitate insertion of our developed technology into MSTAR.

2 Research Questions

In order to achieve the above research objectives, the following specific research questions will be addressed:

1. **Determine a set of model primitives that balance between modeling fidelity and estimation accuracy.** As we mentioned, electromagnetic scattering models are often too detailed to be of practical use for ATR; we will distill from this theory models that contain a few parameters which can be accurately estimated, and which at the same time describe scattering behavior in sufficient detail to effectively discriminate between targets.

   We will characterize achievable estimation accuracy as a function of system parameters, such as bandwidth, center frequency, and signal-t-clutter ratio.

   During the February MSATR interaction meeting, discussions with researchers for the MSTAR Predict module suggest that our proposed model primitive set can be predicted by the MSTAR Predict module. This will facilitate transfer of our research to MSTAR program.

2. **Develop fast, automated computation for complex SAR image-domain processing.** We will develop image domain algorithms because nearly all SAR ATR processing streams operate on image chips. We will develop model order detection methods that are effective for SAR scattering features, to reduce sensitivity of performance on model order. We will develop "hands-off" algorithms that minimize or eliminate user tuning. We will develop algorithms that minimize computational cost while maintaining good statistical feature accuracy.

3. **Implement stand-alone match scoring to evaluate target discriminability and feature estimation tradeoffs.** We will develop a geometric hashing-based algorithm for match scoring. The purpose is twofold: i) to consider feature extraction and feature match scoring in a tight loop to exploit synergy for improved ATR, and ii) to quantitatively evaluate feature extraction performance at the system level as target detection probabilities. While geometric hashing is effective for scattering center locations, a research question is how to extend the procedure for other scattering attributes.

4. **Assess the potential for ATR improvement via superresolution of attributed scattering centers.** Some initial results in superresolution on SAR imagery by Lincoln Labs and ERIM suggest that enhanced resolution improves target discriminability and hence improves ATR performance. Superresolution is an active question of interest in the MSTAR program, to name one. Our attributed scattering center features naturally offer superresolution, both in sub-pixel accuracy of estimated scattering centers, and in ability to extract two (or more) scattering centers within the same resolution "bin". We will explore the gain in performance afforded by superresolution in our attributed scattering center models.

3 Performance Evaluation Process

Our research program includes a plan for quantitatively measuring performance. Below we summarize this plan for the major research goals.

- **Model validation:** The goal is to quantify fidelity of attributed scattering center models. We will use target scattering predictions and measurements from MSTAR of geometric primitive data. We will compare percent energy in our extracted model with original target energy, and also compare detection performance of scattering types for these models.

- **Estimation accuracy:** The goal is to quantify estimation accuracy for feature extraction. We will first use scattering prediction models (MSTAR, XPatch) and measurements of geometric primitives. We will corrupt these data with noise and apply our feature extraction methods. We determine mean and covariance of estimation errors and ROC curves for scattering center detection versus false alarm as SNR varies. We will compare these values with our theoretical predictions of feature uncertainty. We will then validate performance on MSTAR public release SAR imagery. Finally, we will obtain quantitative target classification performance (detection performance ROC curves; confusion matrices) from MSTAR public release imagery when used in conjunction with our match scoring metric.

  A goal of this program is to port our feature extraction algorithms to Khoros modules for use in other programs; thus, another measure of progress is completion of modules.

- **Superresolution:** The goal is to quantify ATR performance improvement afforded by superresolution. We will quantify subpixel accuracy of scattering center location estimates as above. We will also quantify classification performance at the output of our match scoring metric. Using a combination of scattering prediction models at higher resolution and high-resolution measurements as "ground truth", we can quantify attainable sub-pixel accuracy by applying feature extraction on reduced-resolution data.

4 References

References

[Potter and Moses, 1997] Lee C. Potter and Randolph L. Moses. Attributed scattering centers for SAR ATR. *IEEE Transactions on Image Processing*, 6(1):79–91, January 1997.

[Ying, 1996] Chinghui J. Ying. *Stochastic Exponential Modeling and Applications to Radar Signal Processing.* PhD thesis, The Ohio State University, October 1996.

[Chiang, 1996] D.-M. Chiang. *Parametric Signal Processing Techniques for Model Mismatch and Mixed Parameter Estimation.* PhD thesis, The Ohio State University, August 1996.

A Unified, Multiresolution Framework for Automatic Target Recognition

E. Grimson J. Shapiro P. Viola A. Willsky
Massachusetts Institute of Technology Cambridge MA

Abstract

This project focuses on the complete ATR processing chain, from sensor signal processing, to image analysis, to model-based object recognition. It utilizes analysis of performance characteristics of one component to model input characteristics of subsequent components, and leverages the constraints of such analysis to guide the development of the entire chain of target recognition. The overarching theme of our approach is to maintain careful statistical models of each stage in conjunction with multiresolution models and algorithms to derive fundamentally grounded components for all aspects of ATR processing.

1 Introduction

This PI Report describes work conducted under a recently completed ATR URI grant, as well as work that we plan to conduct under a newly issued grant as part of DARPA IU's IMEX/ATR project.

1.1 Objectives – at the top level

Automatic target recognition (ATR) is at a point where it must soon yield significant military advantage. The demands of modern warfare make it a necessity. We can ill afford to wait as our opponents maneuver a SCUD into position or prepare an unexpected attack. We must be able to quickly, automatically and reliably scan an entire theater of operation to detect the whereabouts and intentions of weapons and forces. Unfortunately, even trained operators find target recognition a difficult task. Clearly there is no shortage of information. The quality and quantity of SAR imagery is advancing steadily, as are the vehicles capable of using it, both manned and unmanned. Target recognition is difficult in part because of this unavoidably large volume of data. As a result, information on the target comprises a vanishingly small part of the totality of data, perhaps less than one part in ten million. To make matters worse the image of the target is a highly complex function of the sensor, the measurement scenario and the target itself. ATR is a search for a needle lost in a thousand haystacks— a difficult task made harder because our needle looks a lot like a piece of hay.

Classically, ATR has been formulated as a sequential process. The first step is the processing of raw sensory data, followed by computations that form images from this data. From these images primitive features are detected, and finally a target recognition algorithm is applied to these features. Each step in this process has become a separate area of research, between which there is little communication. For example, SAR based target detection strives to be independent of the processing used to form SAR images from raw data. In our view, revolutionary advances in ATR require more than enhanced algorithms for each of the component functions in an ATR system. Rather, it requires the establishment of a unified theory of ATR, in which the physics of sensor phenomenologies, the tools of statistical estimation, and technology from image understanding (IU) are combined. Such an approach promises to (a) make optimal use of the information embedded in sensor data; (b) yield algorithms that are robust to the uncertainties and variabilities in both phenomena and data; and (c) provide a clear audit trail for the evaluation of performance and for pinpointing the factors limiting performance (e.g., is a better matching algorithm needed or is the problem simply that the data are not of the quality needed to meet specific performance objectives).

Our previous collaborative research under the current ATR-URI program has resulted in a number of accomplishments and outputs along these lines. Among these are a variety of algorithms, several transitions and applications of direct relevance to DARPA programs, and the establishment of very

[0] This report describes research supported in part by ARPA under ONR contract N00014-94-01-0994, by AFOSR grant F49620-93-1-0604, and by DARPA contract 95009-5381. PIs may be contacted at welg@ai.mit.edu, jhs@mit.edu, viola@ai.mit.edu, willsky@mit.edu. URL for the project available at `http://www.ai.mit.edu/projects/darpa/atr/`

close working relationships with a number of key players in ATR programs and ACTD's. However, of equal (and perhaps greater) importance is the fact that our experience in this research effort has both strengthened our belief that an integrated approach to ATR has a considerable amount of promise and also sharpened our understanding of how that promise can be realized. Our research builds on this foundation and offers what we believe is a cohesive set of research ideas that involve the integration of disciplines spanning the entire ATR problem. Perhaps the strongest message that we have extracted from our previous work is that when one takes a unified look at an ATR problem, the challenges presented at each stage of the processing chain is influenced significantly by its placement in context, leading to new problem formulations and conceptual issues that have not been considered before. Our objective is to capitalize on these, both to make significant contributions in domains corresponding to each of these processing steps and to produce new end-to-end processing structures.

1.2 Objectives – detailed examples

While our current research suggests a variety of problems worthy of attention, we can point to two specific contexts that exemplify both the contributions and foundation that our current effort has provided and some very concrete new problem formulations that result from this unified view:

1. Multiresolution SAR-based ATR. One of our major successes to date under the current ATR-URI project is in developing multiresolution stochastic models for SAR imagery that accurately capture the scale-to-scale statistical variability of speckle in SAR imagery. In one application, we used models for natural clutter and for man-made objects together with our fast statistical likelihood calculation methods to develop an enhanced discrimination feature that, when integrated into Lincoln Labs' ATR algorithm and tested on a very large data set, resulted in a factor of 6 reduction in false alarms over the previous best results. In the second application, we used our models to segment natural clutter (trees and grass) and to enhance anomalous pixels (due to man-made scatterers) that did not produce the scale-to-scale variability consistent with natural clutter. The results are very accurate segmentations and enhanced visibility of anomalies as compared to widely used CFAR methods (see Figure 1).

We believe there are many additional applications for these multiscale models. For example, anomalies that result from man-made objects exhibit themselves as distinctive patterns across scale that differ significantly from the scale-to-scale textural variations. Consequently, chains of pixels across scale could in principle be viewed as robust and statistically meaningful *features* that can be further exploited for model-based recognition. We propose to use multiscale features for higher level recognition and reasoning. Classically, target models include geometrical constraints on the appearance of features in space. In this new framework, models will also include information about the appearance of features across scale. The development of such models is a central objective of this project. Once we have such models, we can use our statistically optimal methods for evaluating likelihoods to evaluate match scores for hypothesized models and poses.

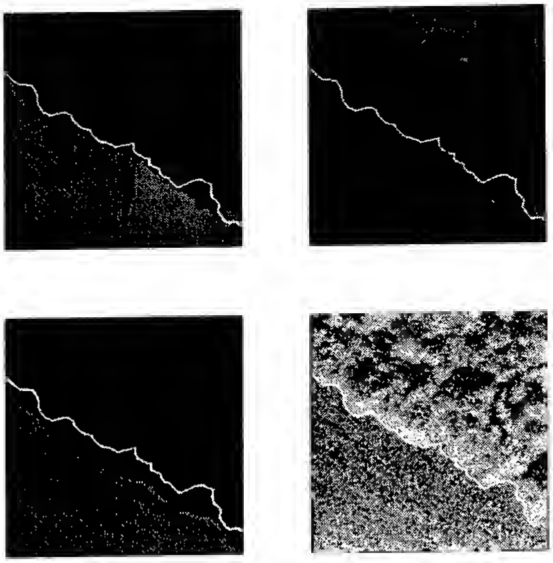


Figure 1: Example of multi-resolution segmentation, by recursively subdividing residual areas based on a multiscale model of statistical variations in the SAR data.

2. SAR-based ATR incorporating pose-dependent SAR image formation and analysis. Scattering patterns for man-made scatterers possess very different characteristics from those of natural scatterers. In particular, while the latter frequently can be modeled as diffuse isotropic scatterers, the former frequently have strong specular characteristics, which implies that they have extremely strong aspect-dependent responses. This effect is even more pronounced in low-frequency SAR as is used for foliage penetration.

While some work has attempted to account for the difference in scatterer type, it is fair to say that a fundamental look at this problem has yet to be taken. Our group has begun to do this, with an initial analysis that indicates that there is significant discriminating information to be obtained if one examines sets of SAR images of a scene constructed using different subapertures of the full SAR aperture. What this suggests is that the natural data structure for higher-level ATR functions will again involve imagery at different resolutions, but that in this case the focus is on exploiting differences in imagery obtained from different parts of the aperture (and hence from different viewing angles). Note that this results in a very novel pose estimation problem: the model of a specular reflector must capture the fact that changes in pose not only change the relative geometry of features but also can change the appearance of these features in imagery from different apertures. As a result, the action of the pose transformation group on a set of images from different apertures and at different resolutions is not simply a geometric transformation.

These two contexts will serve as focal points for our work in the near future. This work, we expect, will build on previous successes we have had in leveraging a cross-disciplinary, statistically sound approach to ATR analysis and development. Examples of our work in this arena are detailed below.

2 Specific Research issues

2.1 Multiresolution methods in laser range imaging

This project addresses laser radar range imaging using realistic models for the uncertainty in the measurements provided by the sensor. Special attention is paid to the so-called range anomalies, which are due to nonlinear combined effect of laser speckle and receiver noise. We use a Haar-wavelet, expectation-maximization (EM) algorithm framework to produce a maximum-likelihood (ML) multiresolution approximation of the range truth. This allows us to tradeoff between the resolution of the reconstruction and the distortions caused by anomalies. That is, as we attempt to resolve finer and finer scale detail we are forced to use a greater percentage of the measured pixels and therefore are constrained in the number that can be declared to be anomalous. Using sensor statistics we can then determine the correct balance between resolution and anomaly rejection.

By exploiting the block-nature of the Haar-wavelet estimation procedure, the maximum-likelihood (ML) fitting, via the expectation-maximization (EM) algorithm, of P Haar-wavelet blocks to a Q-pixel image can be accomplished for practical image sizes with reasonable computational complexity and excellent numerical robustness. This EM/ML processor has proven to be nearly optimal – its estimation performance approaches the ultimate limit set by the complete-data form of the Cramer-Rao bound – with quantifiable performance characteristics.

The fast EM/ML algorithm has been made available for general use. A web page has been mounted *(http://cis.wustl.edu/ mit_cis/ laserradar/ fmlem/ fmlmalg.html)* which provides: the software, a user manual [9], a full technical paper [10], and a sample input file plus processed imagery. Our current research in laser radar range imaging is aimed at wedding the fast EM/ML algorithm to a model-based object recognizer that relies on an EM-based alignment procedure [25]. This effort also leverages off the availability of large amounts of data from the DARPA/MIT Lincoln Laboratory Infrared Airborne Radar Data Release [2].

2.2 Multiresolution models for aspect-dependent SAR imaging

This relatively new effort draws its motivation from earlier work by members of our group and collaborators at Lincoln Labs and Alphatech, which has used Lincoln Laboratory SAR data to demonstrate the efficacy of multiresolution discriminants for distinguishing targets from clutter as well as the benefits of exploiting the aspect-dependence (broadside flash) seen in foliage penetrating SAR imagery. Unlike these efforts, however, the new work builds from fundamental phenomenology originally established for the study of optical SAR [22]. Initial results [17] appear to be quite promising, as highlighted below.

Our work has compared the carrier-to-noise ratio (CNR) signatures produced in an idealized 1-D, continuous-wave, SAR by isolated specular and diffuse reflectors of the same size. Whereas both reflectors produce the same average intensity image, in stripmap operation, when processed to full resolution, such is not the case for multiresolution processing. The coherence of the specular reflector imparts a higher resolution capability – in the intermediate processing regime wherein the along-track chirp compression filter has a processing time that is long enough to produce an effective synthetic aperture, but short enough that the maximum, i.e., transmitter-dwell-limited, synthetic aperture resolution has not been realized – which may provide a powerful discriminant for separating man-made (specular) returns from natural terrain (diffuse) returns. This initial model also provides a theoretical basis for the aspect-dependence (broadside flash) effects expected from a specular target's tilt with respect to the radar-to-ground axis.

We have refined and extended the preceding CNR signatures in several important ways. First, by positing the idealized hypothesis testing problem of detecting a single specular return in the midst of featureless diffuse clutter, we have verified, in this simple case, the essentialy premise stated in the previous paragraph. In particular, he has shown that optimum processing does not employ a maximum-resolution imager. Indeed, under exactly the circumstances that lead to a significant difference in the CNR signatures of specular and diffuse objects, we find that the optimum detection processor has a substantially better detection probability – for the same false-alarm probability – than the full resolution processor. In general, the optimum receiver for this detection problem uses a whitening filter (to flatten the spectrum of the clutter-plus-noise entering the receiver) followed by a matched filter for the whitening-filtered specular return. We have shown, however, that under interesting operating conditions this optimum receiver is well approximated by a simpler multiresolution-filter receiver. The 1-D SAR analysis appears in [18].

Our longer-term plans include: hypothesis testing for a composite target – comprised of multiple target-reflection primitives, i.e., speculars, dihedrals, trihedrals – when it is embedded in diffuse clutter; CNR signatures for stripmap, spotlight-mode, and polarimetric operation of a 2-D SAR. We have already completed full polarimetric generalization of our specular, dihedral, and trihedral target models.

2.3 High-Resolution Pursuit for Robust Feature Extraction

Members of our group have attacked the problem of robust multiresolution feature extraction through a technique which we refer to as high-resolution pursuit. This work, done in collaboration with Prof. Stephane Mallat of Courant Institute, is a variation on Mallat's matching pursuit algorithm involving a new criterion that trades off between global fit and local fit in choosing each of a succession of features. This involves only a modest increase in complexity over matching pursuit but leads to features which are much more clearly connected to physical features in data and which appear to have significant robustness to several types of noise, including additive noise as well as "spiky" noise, as one would expect in speckle-corrupted radar data. The initial application of this method has been to the problem of recognition of objects from their silhouettes. Specifically, there is a natural way in which to map a silhouette into a 1-D signal corresponding to the outline of the object, and a number of recognition methods have been developed that are based on analyzing these 1-D silhouette functions. These methods all have some deficiencies, notably robustness to noise or to the absence of features or the introduction of spurious features due to noise, occlusion, or anomalous estimation of object silhouette as can occur in practice. The vehicle that we are using to test our approach is the recognition of silhouettes of a number of different aircraft, a problem that has been considered by others and which thus provides us with a fair way in which to compare our methods. The results of extensive testing demonstrates the superiority of our new approach, including an increased level of robustness to the types of silhouette errors observed in practice.

This technique has been transitioned to Alphatech, Inc., where high-resolution pursuit is being applied to the problem of robust compression of SAR target models for template-based ATR and to problems of multisensor fusion. In addition, members of our group have recently developed a robust wavelet-based technique for optimal adaptive wavelet representation for noisy data and have begun an examination of the applicability of this method to high-resolution radar feature extraction.

2.4 Multiresolution Analysis of SAR

In earlier work, our group in conjunction with Dr. William Irving (Alphatech), and Dr. Les Novak (Lincoln Labs) had considerable success in constructing and using multiresolution models for SAR imagery as the basis for enhancing Lincoln's discrimination algorithm, resulting in a reduction in false alarm rates by a factor of 6 over Lincoln's baseline algorithm. Following this, our group had extended these multiresolution SAR likelihood ratio methods to the problem of distinguishing different types of natural terrain (trees, grass, etc.). The method we have developed has been shown to provide highly reliable classification decisions and accurate estimates of terrain boundaries. In addition, our group has refined and extended these methods in order to develop efficient SAR compression algorithms that take advantage of the speckle-decorrelating property of our multiresolution models. These methods have been transitioned both to Lincoln and recently to Alphatech, Inc. In addition to these efforts, we have also recently initiated another new project aimed at exploiting the properties of our multiresolution SAR models even further. Specifically, we have begun to look at the use of multiresolution SAR models for optimal or near-optimal model-based ATR. Specifically, since our models do a surprisingly good job of whitening speckle, they can provide the basis for optimal extraction of features corresponding to statistically signficant deviations from this white behavior, which can be directly related to the presence of one or several dominant scatterers. In our work we expect to investigate both enhanced template-based methods as well as fully model-based methods, for which the problems of pose estimation, search, and match take on new twists, thanks to the multiresolution nature of our features. In addition, as an alternative direction we are also exploring the use of multiresolution models to capture aspect-dependent effects of significant scatterers. The key idea here is that the effect of pose in such a situation becomes more complex, since pose can effect not only the location but the appearance of aspect-dependent scatterers.

2.5 Estimation-Theoretic SAR Image Formation for Moving Scenes

Our group, in collaboration with Alphatech, has recently developed the foundation of an estimation-theoretic approach to optimal SAR image formation when there is motion in the scene being imaged. The starting point for this work involves viewing the problem as one of joint position-velocity imaging–i.e., the SAR equivalent of range-Doppler imaging but now in 4 dimensions (2 space and 2 spatial velocity). We have developed a novel and very efficient method for calculating what can alternatively be thought of as the likelihood function for the location in range-rate/cross range of a scatterer or as an image in this 2-D space for each range bin. Using this likelihood function he has now developed a first approach to SAR imaging that does not require motion to be rigid body. Specifically, one of the objectives of this work is to develop a method that can focus an entire SAR image even if different scatterers in the scene are moving differently (including having a target moving across a stationary background). We have recently obtained promising results on simulated data demonstrating that the method can indeed focus images with non-rigid motion. We are also in the process of characterizing the blur in the higher-dimensional imaging framework in order to developed enhanced methods when there are interfering scatterers (e.g. due to background clutter and moving targets). One challenge here is the development of a test environment that is both realistic and that allows in essence Monte Carlo testing in order to obtain statistically significant figures of merit for our approach. The idea here is not to use X-Patch or other simulators but instead to take real SAR data and selectively modify phase histories in order to mimic the effects of motion.

2.6 Segmentation and Feature Extraction of Speckle-Corrupted Imagery

Our group has also had a significant success in a new effort, aimed at developing robust image segmentation algorithms. Our approach is based on a detailed examination of the very active field of nonlinear diffusions and curve evolution in image processing, from which we developed a very simple variant that not only leads to vastly reduced computational burdens but also provides explicit segmentations at a hierarchy of resolutions (rather than requiring substantial postprocessing and interpretation). This new algorithm is characterized by a set of coupled differential equations for the transformation of image pixel values (analogous to the linear differential equations that lead to the scale-space concepts developed by Witkin and others), where in our case the differential equations have a signicant discontinuity that leads both to robust edge identification and to noise removal, including the removal of occasional high-amplitude noise spikes. We have recently demonstrated that this algorithm can perform surprisingly accurate segmentations for the very high speckle levels present in single-polarization SAR imagery. Given this success, we are currently investigating the extension and application of this methodology to problems such as robust feature extraction.

2.7 Mutual Information Based Registration for Recognition and Fusion

Our group has developed a new approach for finding the pose of an object model in an image. In earlier work, we examined methods that utilized the Expectation/Maximization algorithm to trade off solving for the correspondence between model and data features, and solving for the pose of the target in the data coordinate frame. A multiresolution version of the method was designed and implemented to demonstrate the potential efficiencies and robustness gained by using multiresolution data and models. A drawback to this approach is a reliance on explicit uncertainty models. As an alternative, we have developed a new approach for finding the pose of an object model in an image, based on a new formulation of the mutual information between model and image. As applied here the technique is intensity-based, rather than feature-based. It works well in domains where edge or gradient-magnitude based methods have difficulty, yet it is more robust than traditional correlation.

Our approach to aligning the object model to the image is based on the following steps: (1) The mutual information of the model and image is defined, and is expressed in terms of the entropies of several random variables. (2) The entropies and their derivatives are approximated by a method that involves random sampling from the model and image data, or by using histogramming methods. (3) A lo-

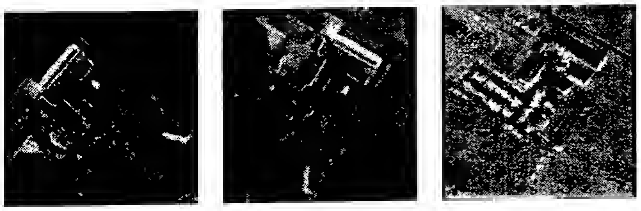


Figure 2: Using Mutual Information to register a SAR image and a visible light image of the same location (from the Stockbridge data set). Left: visible light image. Right: SAR image of same location. Center: the visible light image rotated and translated so that it is aligned with the SAR image.

cal maximum of the mutual information is sought by using a stochastic analog of gradient descent. Steps are repeatedly taken that are proportional to the approximation of the derivative of the mutual information with respect to the transformation.

We have demonstrated the applicability of this method to tracking moving 3D objects in video sequences, to registering SAR with video or other data sources, and to registering target models with extracted SAR data (see Figure 2).

References

[1] J. Baker and W.M. Wells. Multiresolution statistical object recognition. In *Proc. IU Workshop*, 1994.

[2] J.K. Bounds. The infrared airborne radar sensor suite. Technical Report 610, Research Laboratory of Electronics, MIT, (http://cis.wustl.edu/ mit_cis/ laserradar/ IRAR/ IRARmain.html, 1996.

[3] R.D. Chaney, A.S. Willsky, and L.M. Novak. Coherent aspect-dependent sar image formation. In *SPIE Intl Symposium on OE/Aerospace Sensing, Conference on ATR and SAR Processing*, 1994.

[4] R.D. Chaney, A.S. Willsky, and L.M. Novak. Exploiting aspect-coherent imagery for target discrimination in foliage penetrating sar. In *Proc. Joint ATR Systems and Technology Conference IV*, 1994.

[5] M. Daniel and A.S. Willsky. A multiresolution methodology for signal- level fusion and data assimilation with applications in remote sensing. *Proceedings of the IEEE*, 85:164–183, 1997.

[6] C. Fosgate, R. Chaney, A.S. Willsky, H. Krim, W.W. Irving, and W.C. Karl. Multiscale segmentation and anomaly enhancement of sar imagery. *IEEE Trans. on Image Processing*, 6:7–20, 1997.

[7] I. Fung and J.H. Shapiro. Multiresolution laser range profiling. In *Proc. Joint ATR Systems and Technology Conference IV*, 1994.

[8] D.R. Greer. Multiresolution laser radar range profiling of real images. Master's thesis, MIT, 1995.

[9] D.R. Greer. User manual for multiresolution laser radar range profiling – fast ml/em algorithm. Technical report, MIT, 1996.

[10] D.R. Greer, I. Fung, and J.H. Shapiro. Maximum-likelihood multiresolution laser radar range imaging. *IEEE Trans. Image Process.*, 6, 1997.

[11] W.W. Irving. *Stochastic Realization and Identification of Multiresolution Models for Random Processes and Fields.* PhD thesis, MIT, 1995.

[12] W.W. Irving, W.C. Karl, and A.S. Willsky. A statistical procedure for multiresolution model identification and construction. In *IEEE Conference on Decision and Control*, 1994.

[13] W.W. Irving, W.C. Karl, A.S. Willsky, R.D. Chaney, and L.M. Novak. Multiresolution modeling of random fields with application to sar image analysis. In *Proc. Joint ATR Syst. & Tech. Conf. IV*, 1994.

[14] W.W. Irving, A.S. Willsky, and L.M. Novak. A multiresolution approach to discriminating targets from clutter in sar imagery. In *SPIE Symp. on OE/Aerospace Sensing, Conf. on Alg. for SAR II*, 1995.

[15] S. Jaggi. *Multiresolution Feature Extraction Using High-Resolution Pursuit and Morphology.* PhD thesis, MIT, 1997.

[16] S. Jaggi, W.C. Karl, and A.S. Willsky. Multiscale feature extraction and object recognition using wavelets and morphology. In *Intl. Conf. on Image Processing*, 1995.

[17] G. Leung. Synthetic aperture radar discrimination of diffuse and specular target returns. Master's thesis, EECS Dept. MIT, 1997.

[18] G. Leung and J.H. Shapiro. Toward a fundamental understanding of multiresolution sar image formation.

[19] M.R. Luettgen and A.S. Willsky. Multiscale statistical texture discrimination with applications to sar imagery. In *Proc. IEEE Int. Conf. on Image Processing*, 1994.

[20] M.R. Luettgen and A.S. Willsky. Likelihood calculation for multiscale image models with applications in texture discrimination. *IEEE Trans. on Image Process.*, 4:194–207, 1995.

[21] E.L. Miller and A.S. Willsky. A multiscale approach to sensor fusion and the solution of linear inverse problems. *Journal on Applied and Computational Harmonic Analysis*, 2:127–147, 1995.

[22] D. Park and J.H. Shapiro. Performance analysis of optical synthetic aperture radars. *Proc. SPIE*, 999:100–116, 1989.

[23] P. Viola. *Entropy, Information, Computer Vision and Image Processing.* PhD thesis, MIT, 1995.

[24] P. Viola and W. Wells. Alignment by maximization of mutual information. In *Fifth Intl Conference on Computer Vision*, 1995.

[25] W.M. Wells. Statistical object recognition. Technical Report 1398, Artificial Intelligence Laboratory, MIT, 1993.

Statistical Independent and Relevant Feature Extraction for Classification of SAR Imagery

Jose C. Principe, Ph.D.
Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL 32611
principe@cnel.ufl.edu, http://www.cnel.ufl.edu

1. Motivation

The first topic of the research deals with the problem of automatically extracting features from imagery. The selection of features has been primarily an heuristic procedure. Humans pick features that seem to be relevant to solve the problem based on their experience with the data. This procedure has two shortcomings: one does not know a priori the discriminant power of the features, and second, when the signals (or images) are collected with new sensors or the imagery is new there is no available knowledge so there is a lag in the exploitation of the imagery. We believe that this may be happening in synthetic aperture radar (SAR) imagery due to the novelty of the technique and the principles involved in image formation, which are different from optical images (Figure 1).

Figure 1. SAR data of trees, a house and a water tower (MIT/LL mission 90). Where is the water tower?

Our approach seeks to develop a new methodology to extract automatically features from the imagery using information theoretic principles. Our ultimate goal is to perform feature extraction through optimization of the information transfer from the original image to a subspace. If we are successful, a general and systematic procedure will be available to extract features from any type of imagery. Since there is no desired response at the feature extraction stage, the method has to be self-organizing.

The second aspect of the research is to enhance our present methodology of training classifiers. So far our classifiers are trained in the laboratory and their parameters set "for ever". There is strong evidence that the performance of any machine that learns from the environment (i.e. through induction) is ultimately limited by the amount of data that it is trained with. This means that machines should always be learning when exposed to new data, even after being deployed. There is little knowledge how to do this in a systematic and robust way. So a second goal of this research is to seek methods to continue training classifiers after deployment.

2. Research Questions

a) Statistical Independent features

Formulate feature extraction as a process of transferring the most information about the input signal through a special information channel which includes a projection to a subspace.

- Develop an algorithmic methodology to implement the feature extraction.

Acknowledgment: This work was partially support by DARPA grant F33615-97-1019.

- Characterize the methodology and compare it to independent component analysis and principal component analysis.
- Validate the method with SAR data and compare it with other existing methods of feature extraction.

b) Training after deployment

- Seek a methodology to keep training classifiers after deployment by using ideas from reinforcement learning.
- Study alternate methodologies based on the EM algorithm and the newly introduced mixture of experts model.

3. Relevance

This work is relevant because it is addressing fundamental questions in pattern recognition. It will also impact the present state-of-the-art in automatic target recognition (ATR) using SAR. SAR is a new imagery that is very different from optical images. What are the most discriminant features in SAR? How shall we extract them? What is the information content of SAR versus optical imagery? These are very important questions that need to be answered using a principled approach as the one to be developed here.

The second aspect is also very important for ATR since these systems are a mixture of model based approaches with data driven methods (see the MSTAR program). We firmly believe that sooner or later the performance limit will be the lack of data used to train these systems. Data collection is extremely time consuming and expensive. However, ATR systems are exposed to lots of data when they are deployed. The issue is how to harness this exposure to new data and keep training the ATR system.

4. Methodologies

In the first topic (feature extraction) we plan to use information theory and artificial neural networks to create nonlinear subspace projections that preserve as much as possible the input image information after the projection into the feature space. We have already developed a methodology to manipulate the information in the output We still need robust training algorithms to perform the task. The characterization of the quality of the method and comparison with alternate methodologies is also necessary.

In the second topic (training after deployment), we will formulate the problem using reinforcement learning algorithms to include the scalar input from the operator (Yes/No). We will also study other self-organizing alternatives such as the EM (estimation maximization) algorithm and the recently introduced mixture of expert (MOE) model (see our paper in the proceedings).

Please see our WEB page for further details (http://www.cnel.ufl.edu/).

5. Evaluation

We will start with some easy to interpret "synthetic" cases to obtain a better understanding of the algorithms, but the bulk of the evaluation will utilize the newly released SAR data for the MSTAR program. Our goal is to compare the performance of our feature extractor with more traditional features such as the gamma intensity features [1], [2]. We propose to use the same classifier (such as the nonlinear MACE filter [3] and the MSTAR classifier stage) and substitute our features with the traditionally utilized for perfromance comparisons. The comparison will follow the well established procedure of ROC (receiver operating characteristics) curves. We expect to provide technology transfer to MSTAR contractors developing focus of attention and segmentation algorithms. They will be the ultimate judges of the quality of our work. Part of our technology transfer is also to develop modules for the image understanding environment.

6. References

Principe J., Radisavljevic A., Kim M., Fisher J., Hyett M., Novak L., "Target presecreening based on 2D gamma kernels", in Proc. SPIE 95 Conference, vol. 2487, pp 251-8. Orlando, Florida, 1995.

Kim M., Principe J., "A New CFAR Stencil for Target Detection in Syntehtic Aperture Radar Imagery", SPIE 96, Orlando.

Fisher J., Principe J., "A nonlinear extension to the MACE filter", Neural Networks, vol. 8, #17/8, 1131 -1141, 1995.

Context and Quasi-Invariants in ATR with SAR Imagery

Thomas O. Binford, By-Her Wang, and Tod S. Levitt *

Robotics Laboratory, Computer Science Department
Stanford University, Stanford, CA 94305
http://robotics.stanford.edu/groups/reality/binford.html

Abstract

This analysis of XPatch synthesized SAR images shows promising results on existence of persistent scatterers and their measurement in XPatch imagery. Preliminary results based on persistent scatterers and a generic vehicle model demonstrate estimation of vehicle orientation to about 3 degrees, length between wheels to about 30 cm, and estimating number of wheels, without knowledge of vehicle class or pose. Planned research will investigate recognition using generic target and clutter models in analysis of a stable, useful subset of radar images. Research will include quantitative characterization of persistent scatterers and stable relations, an end-to-end recognition test, and investigation of supporting technology for MSTAR. The paradigm offers a genuine methodology to cope with articulation and obscuration.

1 Introduction

SAR imagery is known to vary rapidly with target angle. Scatterers appear and disappear over a few degrees. The resultant image variability makes ATR template matching computationally complex, requiring search for match over a relatively dense set of angles. That image variability also lessens the discrimination power of template matching in ATR, because data images cannot be guaranteed to match exactly. Matching errors tend to introduce biases in matching; matching algorithms must relax matching constraints. Existing systems work reasonably for unarticulated, unobstructed targets, *e.g.* MSTAR and [Ikeuchi 96; Novak 94; Novak 95; Novak 96]. However, results degrade with obscuration and articulation.

Research leading to this project was intended to investigate whether a stable, useful subset of radar images could be found, i.e. to investigate whether persistent scatterers exist, enough scatterers sufficiently persistent that indexing could be done initially with persistent scatterers, to characterize their behavior in a preliminary way, and to demonstrate that scatterer peaks could be estimated accurately to represent target signals in MSTAR or in an intended matching algorithm.

We were motivated by analysis of scattering of target components to ask whether stable scatterers exist. A component scattering model was not regarded as plausible in 1991. In turntable experiments, [Dudgeon 94] found evidence for persistent scatterers, a minimum of about 10 at any angle, persistent for a minimum of 20 degrees, with no disappearance of more than one degree. Subsequently, we found persistent scatterers in XPatch data. A better analytic understanding was since derived for persistent scatterers and for a generic component scattering model.

An analysis based on partial matching of persistent features to generic vehicle models was proposed as a solution to recognition with obscuration and articulation. Generic models with partial matching is also a methodology for hypothesis generation without exhaustive enumeration of targets and poses. Lessening variability by choosing persistent scatterers and stable relations might possibly compensate in part for the loss of unavoidable loss of discriminability with obscuration and articulation.

The objective of this project is to develop target recognition based on persistent scatterers and stable relations in the support of the MSTAR paradigm for ATR and in development of a new recognition system. The following subgoals lie along the path toward that goal: to characterize persistent scatterers

*This research was supported by a contract from the Air Force, F33615-93-1-1281 through WPAFB from ARPA ASTO "Multi-Sensor ATR: Quasi-Invariants and High Accuracy Measurements in Bayesian Inference" and "Context and Quasi-Invariants in ATR with SAR Imagery".

further; to characterize stable relations among persistent scatterers; to investigate non-persistent scatterers and incorporate matching them in a matching scheme; to develop a generic image model and generic target model; to derive and implement a recognition algorithm in a Bayesian network [Binford *et. al.* 87], based on generic image and target models to estimate angle and dimensions of vehicles without enumerating vehicle class and pose.

At the same time, context is widely regarded as an effective way to eliminate large classes of clutter from fixed targets and civilian vehicles [Levitt 95]. This research will investigate context-enhanced discrimination of targets vs natural and cultural clutter based on correspondence between context from available databases (*e.g.* DMIF or HUB) and extended structures registered from imagery [Oliver 90; Posner 93; Wellman 93]. Progress in using context will come from improved discrimination using structures found from texture analysis and from buildings discriminated from analysis of leading edges in their radar image.

2 Toward an Effective Methodology

Some questions need to be answered concerning effectiveness of this methodology. 1) Is there a useful, stable subset of a SAR image, enough persistent scatterers to constrain target hypotheses, sufficiently persistent to implement a hierarchical indexing scheme proposed in this component-based, body-centered recognition paradigm? 2) Are there stable relations among persistent scatterers that enable partial matching that is computationally affordable? Among partial matching algorithms are both computationally effective and computationally expensive algorithms. 3) Can features be extracted from persistent scatterers accurately enough over a broad range of target orientations with enough independence from interference among scatterers to enable stable estimation of scatterer locations and stable estimation of relations among them to enable estimating vehicle orientation and dimensions without knowing vehicle class or orientation. 4) Can a generic vehicle model be formulated that gives a basis for predicting automatically which stable relations will exist and be useful in a broad range of images.

We show preliminary results for XPatch data for three vehicles. Using synthesized data provides data over a range of controlled situations essential to developing a sound algorithm basis, before development on real data. Ikeuchi at CMU provided the XPatch data. In Figure 1, at the top is a low quality optical image of a BTR60. Below it are 6 SAR images from XPatch at a range of angles: 6, 12, 18 degrees, then below, 30, 36, 42 degrees. Note that the SAR images are defocussed in the range direction; presumably they could be focussed better by a choice of XPatch parameters. At the bottom are intensity profiles along the leading edge of the target at angles of 31, 37, and 43 degrees and 69, 75, and 81 degrees. We see four peaks with nearly constant amplitude (to 15%). Note that the peaks persist from 6 degrees to 174 degrees, nearly the whole target angle range. The BTR60 has four wheels with wheel spacing consistent with the good measurements available from peak detection. Wheel spacings provide a geometric invariant (or several) and several geometric quasi-invariants. Peak amplitudes provide photometric quasi-invariants independent of photometric calibration of the image.

Figure 2 shows evidence of persistent scatterers over the whole vehicle. In the upper left are peak scatterers for the BTR60, extracted by our peak detection algorithms from XPatch images over angles from 45 to 135 degrees. Results are similar over nearly 0 to 180 degrees. Peaks extracted from XPatch images at 1 degree intervals for constant slant angle are rotated to zero degrees azimuth and superimposed. We see four clear clusters on the leading edge and eight or nine more clusters interior on the vehicle, along the top. On the middle right, the four peaks were isolated using their statistical distribution. In the lower left, the four peaks are shown in a 3D plot intended to demonstrate in an intuitive and semi-quantitative way the level of persistence of the scatterers. The plot shows (x,y) position as a function of azimuth angle at angles from 45 to 135 degrees. If we look along the vehicle leading edge (x), we see that the peaks are well-separated, with few dropouts.

Existence of stable relations seem to be born out by measurements of ratios of wheel spacings over large ranges of angles. There is also strong analytic reasoning in support of geometric invariants and quasi-invariants. Subsequent detailed SAR image analysis may strengthen and refine this expectation. Much more analysis will be made of similar data using XPatch backtrace facilities to determine where the true scatterers are, to quantify variability of individual scatterers for incorporation into estimation and decision algorithms, and to serve as ground truth for peak estimation and feature detection. Non-persistent scatterers and the ratio of persistent to non-persistent will be studied. These investigations will be carried out in collaboration with Vince Velten from Wright-Patterson AFB and Dr. Eamon Barret and Dr. Paul Payton from Lockheed-Martin.

Figure 3 shows preliminary steps in target recognition. In the upper left is an XPatch image of the BTR60 at 45 degrees. The lower left shows the Delaunay triangulation of peaks detected from the XPatch image in the target cluster. In the upper right are peaks detected in the image, with peaks on the longer leading edge marked with a surrounding circle; peaks on the shorter leading edge are marked with a cross. The leading edge perpendicular pair is shown by solid lines. In the lower right, candidate peaks along the leading edge were selected for intensity from the generic vehicle model to generate candidates for wheels. Estimated leading edge orientation, length, width, and distances between wheels are shown.

We are developing an algorithm for estimating leading edges of targets, insensitive to points not on the leading edge. This is a problem for which least-squares estimation is ill-suited. Least squares weights most heavily points that are furthest away. A non least-squares method was developed, essentially that used in linking in the Binford-Horn edge operator. That operator will be used in computing leading edges of buildings to discriminate buildings and other fixed clutter from targets.

Figure 4 shows estimation of azimuth angles over a broad range of angles in XPatch data for three targets, BTR60, KTANK, and BMP. These estimates cover a range from 10 to 170 degrees; above we saw that the range could be extended a little further, but it cannot be extended below about 5 degrees. [Dudgeon 94] observed that zero degrees and ninety degrees were special views. Those views showed up in our experiments and for physical reasons, in our generic vehicle model.

Over this wide range of angles, the standard deviations for vehicle azimuth angles were 4.0, 2.3, and 3.7 degrees. We believe that the measurement will improve because we are still developing the algorithm. It still makes mistakes in including interior, non leading edge points that bias the result. Standard deviations are strongly affected by these mistakes, i.e. wild points.

Figure 5 shows persistence for KTANK similar to Figure 2 for the BTR60. In Figure 5, top, peaks from views at 1 degree intervals from 0 to 180 degrees were rotated into the 0 degree coordinate frame and superimposed. At the bottom are peaks from the longer leading edge Vive peaks appear more irregular than in the BTR60 images. Again, the peaks persist over nearly the whole angle range.

Similar results were obtained for the BMP. They are omitted to cut space and to avoid detail.

3 Observations

Persistent scatterers have been demonstrated in real SAR from turntable data [Dudgeon 94], in XPatch synthesized SAR, and predicted in analysis. At this early stage, the apparent ratio of persistent to non-persistent scatterers is more favorable than expected. There appear to be about a dozen persistent scatterers on vehicles like the three shown here. Their persistence appears over a broader range of angles than initially expected but now appears to follow from good reasons.

Are there enough scatterers? There appear to be about 12 persistent scatterers typical for these targets. If only 30% of them were obscured, the remainder seem sufficient. In SAR, two scatterers along the leading edge provides a measured constraint; three scatterers are over-constrained. Relative to a leading edge estimate, an additional point off the leading edge is another constraint on matching. With only three points, there are two constraints.

A next step is incorporating points interior to the leading edge, again exploiting the generic model. Some of that can still be done at a generic level for vehicles.

One foot resolution appears to be a magic resolution at which useful detail becomes apparent for vehicles the size of the military vehicles used in these XPatch studies, about 8 meters, 27 feet. Although many pixels cover these vehicles' images, it is the size of scatterers that matters here. There are few pixels on wheels, turrets, and other observable structures. Our estimation methods seem to be on the borderline of resolving the wheels on the vehicles. Making effective use of available image resolution is extemely important.

It seems that improvements in the peak detection provides strong leverage for the problem. I.e. resolving the wheels in all cases on the leading edge would contribute substantially to classifying vehicles. The current peak detection was conceived for isolated peaks; it appears very useful in terrain and vegetation. On targets, peaks overlap, affecting peak detection. Rather than implement our understanding of a solution to the overlapping peak roblem, we have chosen thus far to push on to a broader, birds-eye view of the issues. We are considering improvements in image and in feature detection in the raw SAR phase history data before image formation or exploiting those advances of others in superresolution [Cabrera 94; Mann 92; Odendaal 94].

In fact, measurement of dimensions seems to be relatively accurate. The distance between wheels has a standard deviation of 10 inches. These dimensions go a long way toward constraining vehicle class.

4 Conclusions

These preliminary results appear to warrant further investigation. Persistent scatterers have been demonstrated in real and synthesized SAR experiments; they are expected to exist for good reasons. At this early stage, persistence appears over nearly the whole angle range, again for good reasons. A sufficient number of persistent scatterers appear that it seems promising to recognition. Peak detection seems adequate for initial investigation. further accuracy refinement may be necessary to exploit available imagery. A generic vehicle model has proved surprisingly powerful.

5 References

[Benitz 94] G.R.Benitz, "Adaptive High-Definition Imaging"; *SPIE* Vol 2230, pp 106-119, 1994.

[Binford 87] T.O.Binford, *et. al.*, "Bayesian Inference in Model-Based Machine Vision"; *Proc Workshop on Uncertainty in AI*, AAAI87, 1987.

[Cabrera 94] S.D.Cabrera, *et. al.*, "Application of One-Dimensional Adaptive Extrapolation to Improve Resolution in Range-Doppler Imaging"; *SPIE* Vol 2230, pp 135-145, 1994.

[Dudgeon 94] D.E.Dudgeon, *et. al.*, "Use of persistent scatterers for model-based recognition"; *SPIE* Vol 2230, pp 356-368, 1994.

[Haag 91] N.N.Haag, *et. al.*, "Invariant Relationships in Side-Looking Synthetic Aperture Imagery"; *Photgrammetric Engineering and Remote Sensing*, V 57 pp 927-931, 1991.

[Ikeuchi 96] K.Ikeuchi, *et. al.*, "Invariant Histograms and Deformable Template Matching for SAR Target Recognition"; *Proc IEEE CVPR*, pp 100-105, 1996.

[Jane 84] Foss, Christopher F. "Jane's Light Tanks and Armoured Cars"; 1984.

[Levitt 95] Levitt, T.S., *et. al.*,"Bayesian Inference-Based Fusion of Radar Imagery, Military Forces and Tactical Terrain Models in the Image Exploitation System/Balanced Technology Initiative", *Intl. J. of Human-Computer Studies*, No. 42, 1995.

[Mann 92] J.Mann and R.Hummel, "Synthetic Aperture Radar without Fourier Transforms"; Courant Institute, 1992.

[Moulin 93] P.Moulin, "A Wavelet Regularization Method for Diffuse Radar-Target Imaging and Speckle-Noise Reduction"; *Journal of Mathematical Imaging and Vision*, 3, 123-134, 1993.

[Novak 94] L.M.Novak and G.J.Owirka, "Radar Target Identification Using an Eigen-Image Approach"; *IEEE National Radar Conference*, Atlanta, GA, 1994.

[Novak 95] L.M.Novak, *et. al.*, "Effects of Polarization and Resolution on the Performance of a SAR Automatic Target Recognition System"; *Lincoln Laboratory Journal*, Vol 8, pp 49-68, 1995.

[Novak 96] L.M.Novak, *et. al.*, "ATR Performance Using Enhanced Resolution ATR"; *SPIE Conf on Algorithms for Synthetic Aperture Radar Imagery III*, 1996.

[Odendaal 94] J.W.Odendaal, *et. al.*, "Two-Dimensional Super resoltuion Radar Using the MUSIC Algorithm"; *IEEE Transactions on Antennas and Propagation*, V 42, pp 1386-1391, 1994.

[Oliver 90] C.J.Oliver, "Clutter Classification based on a correlated noise model"; *Inverse Problems*, 6, PP 77-89, 1990.

[Owirka 94] G.J.Owirka and L.M.Novak, "A New SAR ATR Algorithm Suite"; *Conf on Algorithms for Synthetic Aperture Radar*, 1994.

[Posner 93] F.L.Posner, "Texture and Speckle in High Resolution Synthetic Aperture Radar Clutter"; *IEEE Trans on Geoscience and Remote Sensing*, V 31, pp 192-203, 1993.

[Wang 96] B. Wang and T.O. Binford, "Generic, Model-based Estimation and Detection of Peaks in Image Surfaces", *Proceedings of Image Understanding Workshop*, Vol. 2, pp.913-922, Feb. 1996.

[Wellman93] R. J. Wellman, *et. al.*, "Radar Cross Sections of Ground Clutter at 95 GHz for Summer and Fall Conditions"; AGARD meeting on Atmospheric Propagation Effects through Natural and Man-Made Obscurants for Visible to MM-Wae Radiation, 1993.

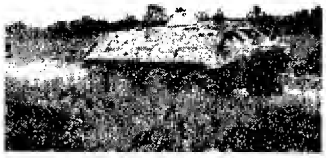
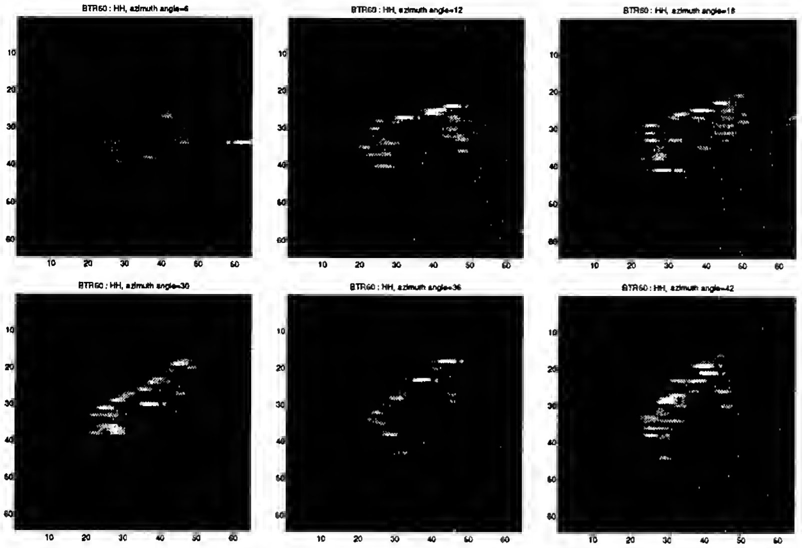
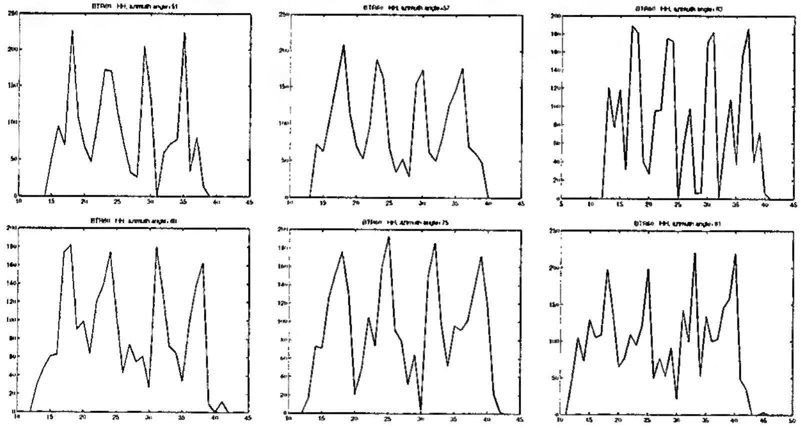


Figure 1: BTR60: Right Front



BTR60 : HH, azimuth angle=6
BTR60 : HH, azimuth angle=12
BTR60 : HH, azimuth angle=18
BTR60 : HH, azimuth angle=30
BTR60 : HH, azimuth angle=36
BTR60 : HH, azimuth angle=42
10
20
30
40
50
60

XPATCH DATA: azimuth angle= 6,12,18,30,36 and 42



Profiles of Four wheels

Figure 1: A BTR60, its XPatch images, and intensity profiles along its leading edge; (a) an optical image of a BTR60; (b) XPatch images of the BTR60 over a range of angles; (c) intensity profiles, a slice through the XPatch image along the leading edge of the target.

EXAMPLE OF PERSISTENT SCATTERERS: BTR60

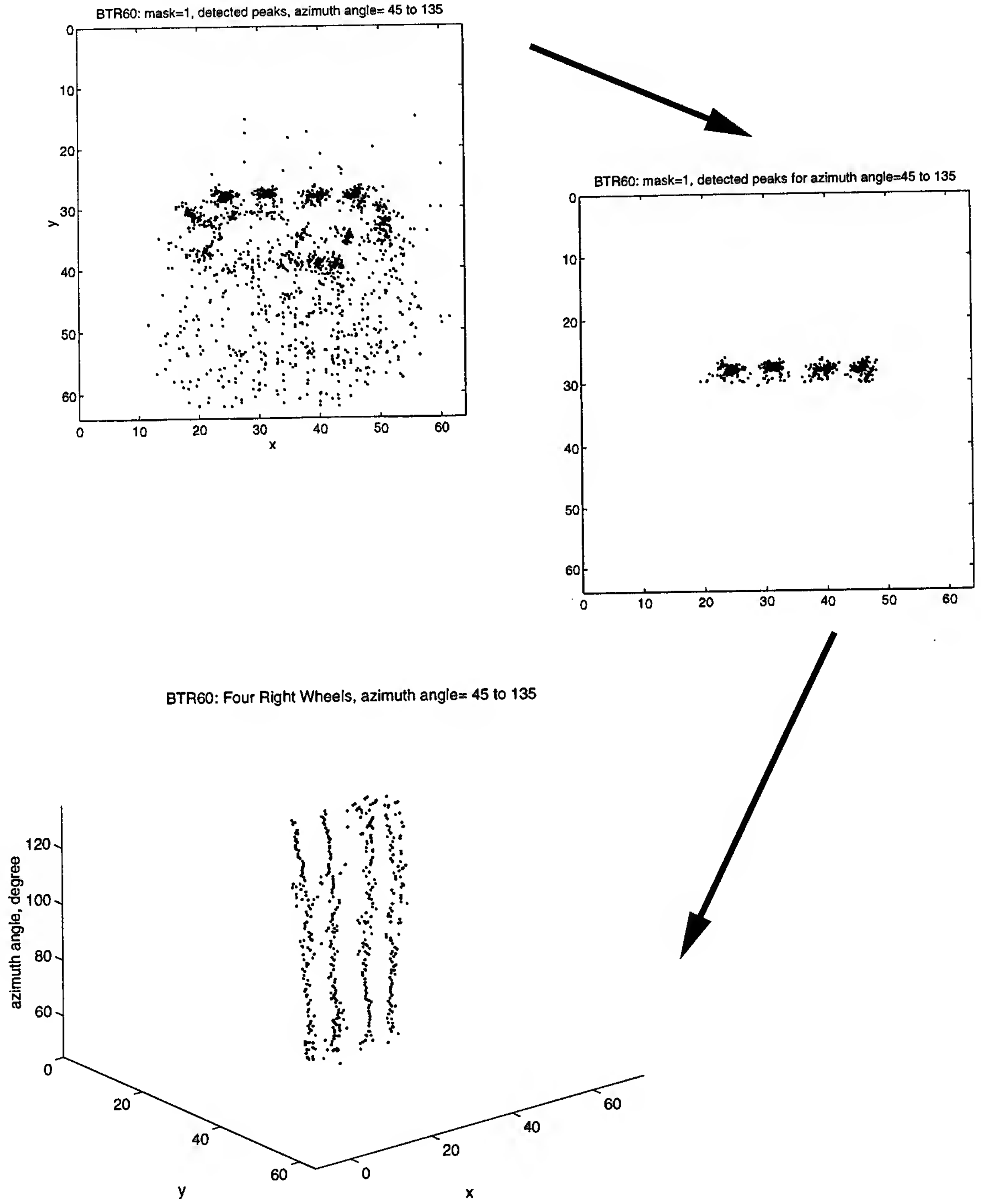
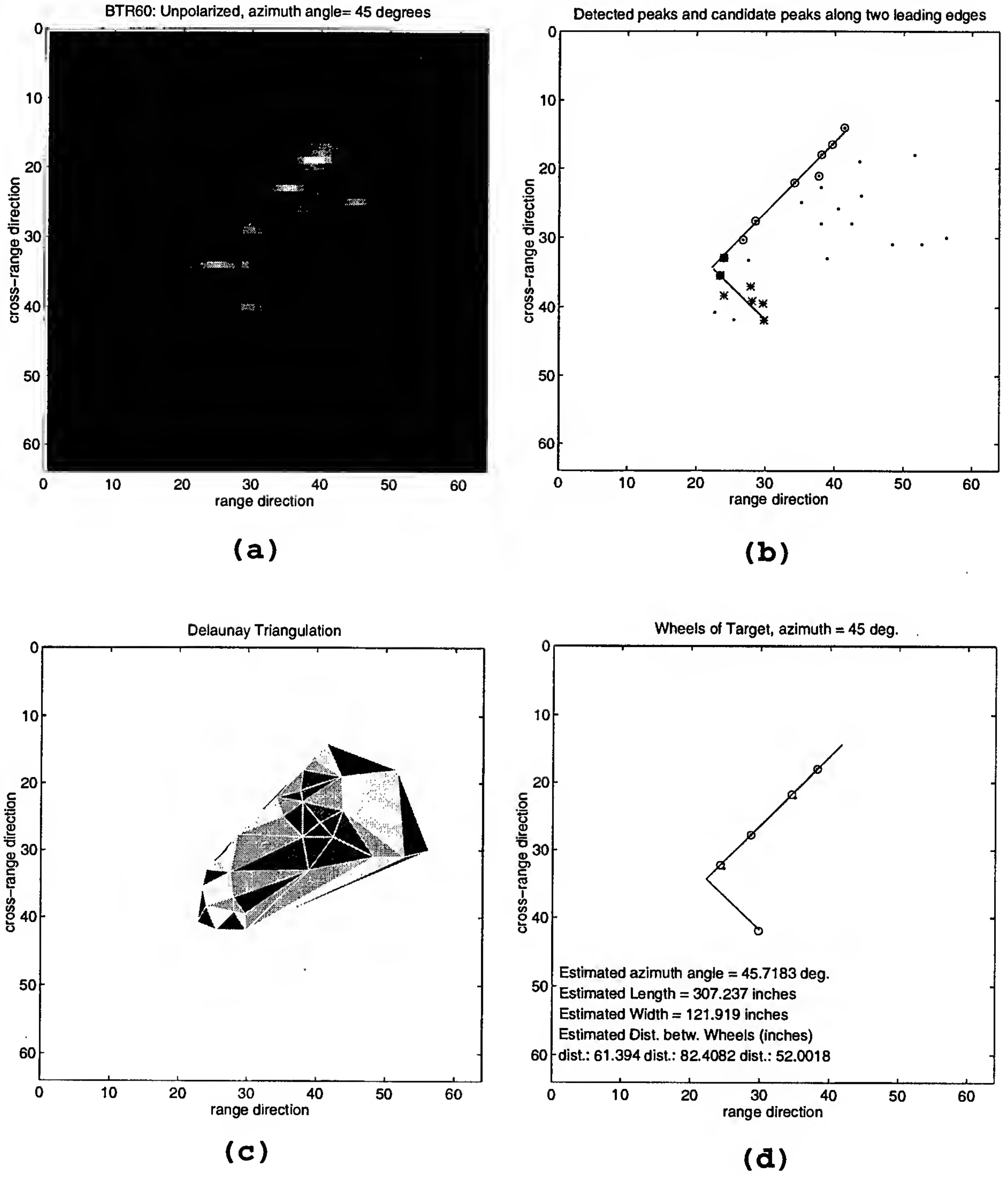


Figure 2: (a) For a BTR60, peaks for view angles differing by 1 degree are rotated and superimposed; (b) clusters around the wheels are isolated; (c) a 3d plot of clusters around wheel in (x,y) with azimuth along the z axis.

Target Recognition

Example: BTR60



(a)

(b)

(c)

(d)

Figure 3: (a) The XPatch image for a BTR60 at 45 degrees; (b) the Delaunay triangulation establishes relations among peaks in the target cluster; c) peaks from the target with the perpendicular leading edge pair superimposed; candidates for the leading edges are marked with circles for the longer leading edge, with crosses for the shorter; (d) peaks that lie along the leading edge are selected on intensity; ratios of wheel spacings match those of the BTR60.

Estimations of Azimuth Angles:XPATCH DATA

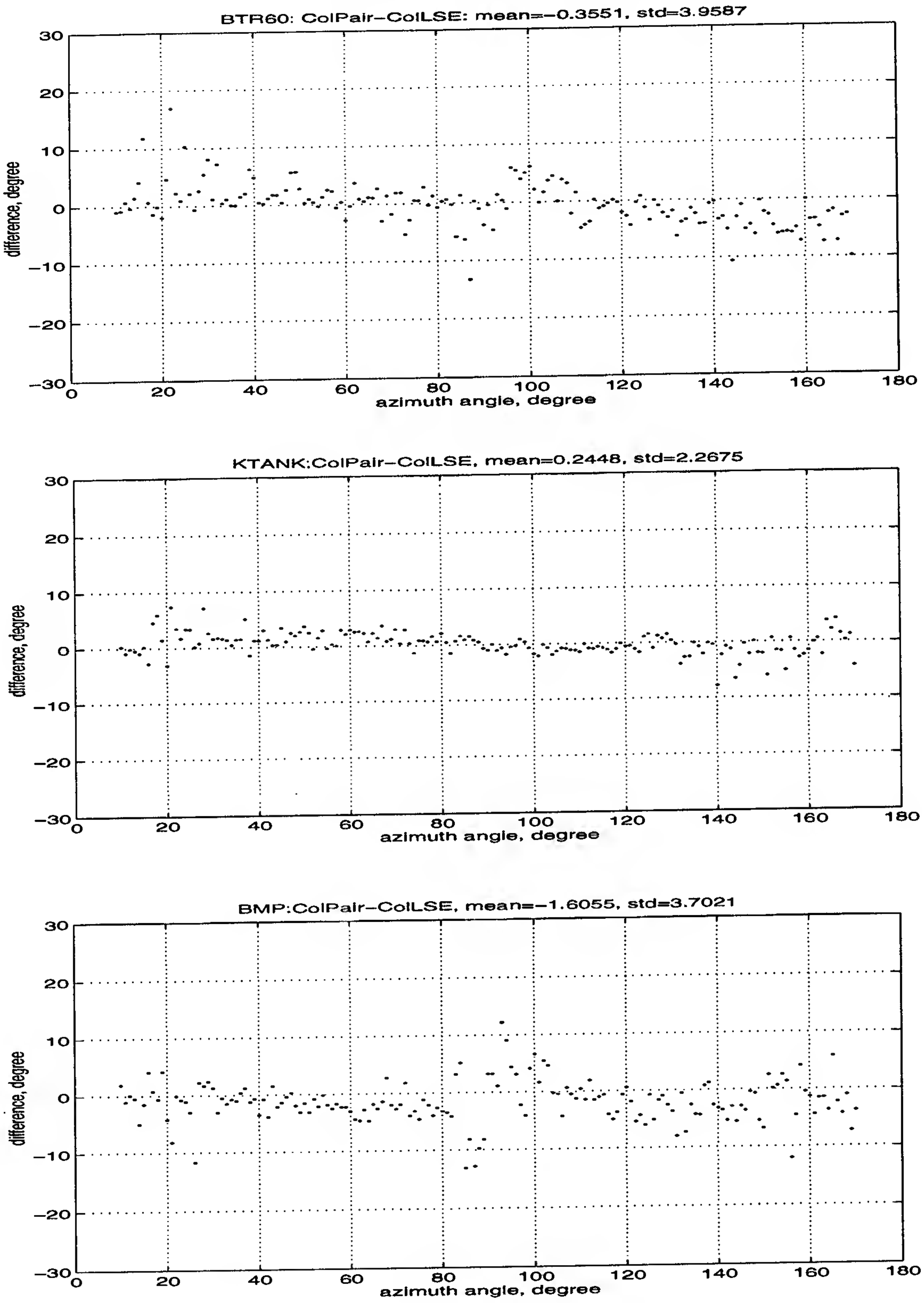


Figure 4: Estimates of azimuth of three targets, BTR60, KTANK, and BMP over 10 to 170 degrees.

Persistent Scatterers: KTANK

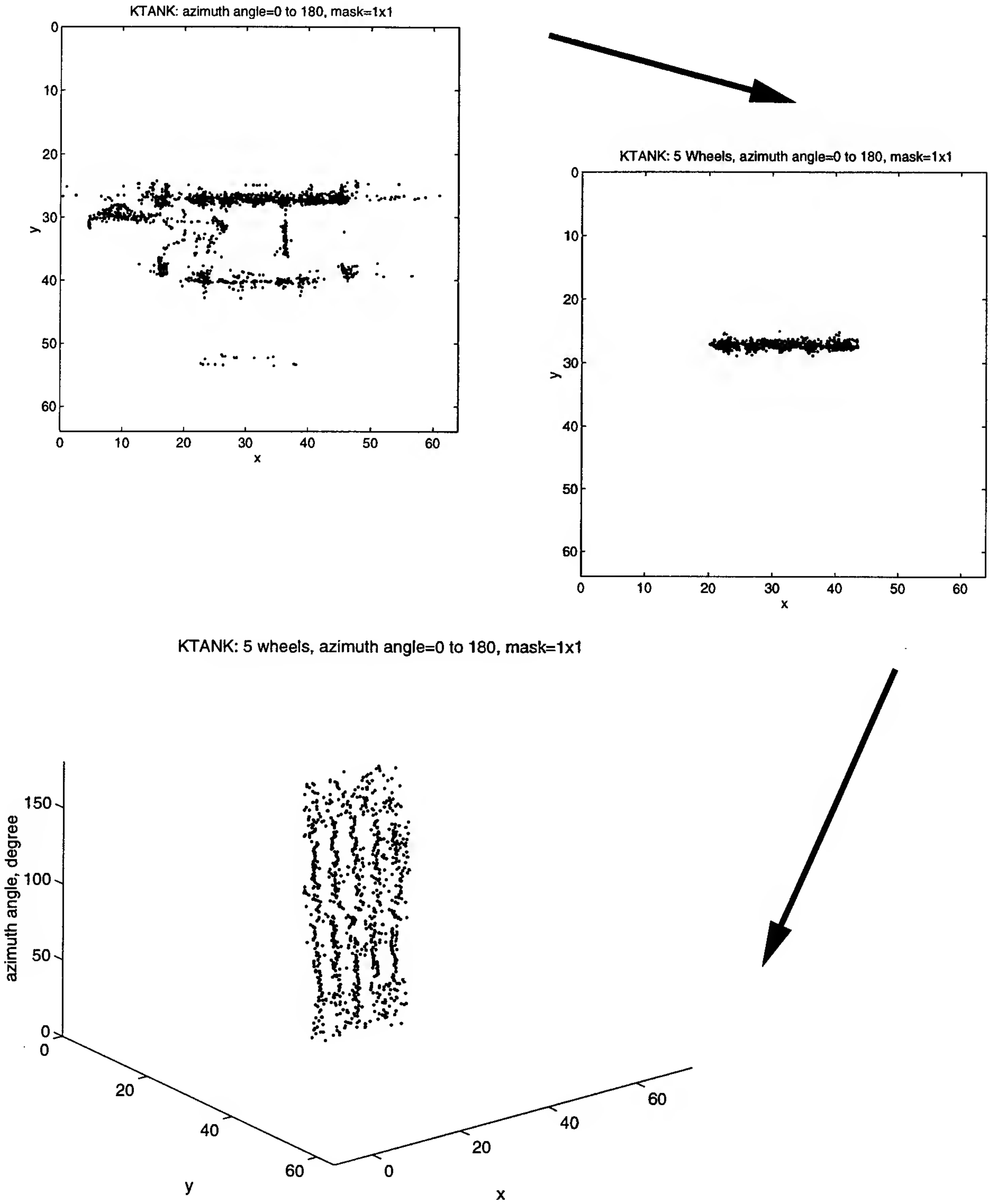


Figure 5: (a) For a KTANK, peaks for view angles differing by 1 degree are rotated and superimposed; (b) clusters around the wheels are isolated; (c) a 3d plot of clusters around wheel in (x,y) with azimuth along the z axis.

3D Object Recognition from Multiple and Single Views

Isaac Weiss **Azriel Rosenfeld**
Center for Automation Research, University of Maryland
College Park, MD 20742-3275 (weiss@cfar.umd.edu)

Abstract

The goal of the project described here is to perform recognition of 3D objects in cluttered, unstructured environments. The objects (such as targets) can be partially occluded. Such unconstrained object recognition involves high computational complexity, and we will apply new methods to drastically reduce this complexity. These methods include: i) Use of invariant constraints. Invariance has been proven successful for planar objects, but until recently it could not be applied to images of general 3D objects. ii) Use of curve features such as (non-coplanar) conics rather than point features. The system developed will work in two main modes: i) multiple views; ii) a single view plus modeling, such as a known data-base or 3D skew-symmetry. Reliability will be of major concern in both modes.

1 Introduction

We describe here a new concept for a system that detects and identifies objects, either from a single image or from two or more images. The objects will be recognized in an unconstrained and unstructured environment. The system will be tested on real imagery available to the project.

Recognizing objects has been a major goal of IU research, but major obstacles have been encountered. Among them:

(i) Complexity. In a typical recognition system, an observed object is compared to a database of models. Recognition means a positive match between the object and one of the models. Matching of visual data is very costly computationally, unlike matching of alphanumeric data. Given a large number of objects, and a large database of models, we face a difficult "high complexity" problem of comparing each observed object to each model. The complexity increases greatly when the number of visible objects and models is increased. Currently this poses a severe limit on the the number of objects that can be recognized. Using a new mathematical theory, we will reduce the complexity by orders of magnitude, increasing the capabilities of the system by a factor of hundreds or thousands.

(ii) Viewpoint dependency. Compounding the problem of complexity is the fact that the observed image of an object depends on the point of view from which the object is observed. We would like to store in a database only one model of each object. Several current systems use viewpoint-invariant descriptors, i.e. they extract from the observed object some descriptors which are invariant to the viewpoint and compare them to similar descriptors stored in a database. However, these systems are limited to objects that are planar or have mostly planar contours. The proposed system will deal invariantly with objects of arbitrary three-dimensional shapes.

(iii) Projection. When a three-dimensional object is projected into a two-dimensional image, "depth" information is lost. Two objects which look the same in the image can be, in principle, very different in the 3D reality. This can be solved by using multiple images, but then we again encounter the complexity problem as we try to match features of the different views. Here again we increase the efficiency relative to previous methods by orders of magnitude.

(iv) Reliability. This is a major problem in current systems. Most measurements on real images are subject to significant noise and errors. We will deal with this in several ways. One way is cross-verification.

This project will be supported by the Defense Advanced Research Projects Agency (ARPA Order No. E655) and the U. S. Air Force Wright Laboratory. For further information see
http://www.cfar.umd.edu/~weiss/atr.html .

When observing an object such as an airplane, we will not be satisfied with matching it with a model airplane in the database. We will also want to make definite identifications of parts of the object, such as wings or engines. Only when several levels of parts and subparts of the object have been identified will we make a positive identification of the whole object. This is made possible by our capability discussed earlier, namely that we can identify many more objects, and therefore sub-objects, than current systems.

Another way to increase reliability is to use more reliable features extracted from the object. Most systems rely on distinctive points or lines observed on the objects. In addition to these we will also use conics such as circular or elliptical parts of the object. These are usually more reliable since each conic consists of many points, so that the errors associated with measuring each point tend to average out. The use of conics will also reduce the complexity, because there are far fewer of them than there are points. Previous methods used only coplanar conics. We will use conics in general 3D configurations.

Previously used invariant-based methods have suffered from the fact that the invariant quantities are sometimes susceptible to errors. To overcome this problem, in addition to the above methods, we will include in our algorithm a final matching and verification stage which is independent of invariants.

2 Technical Approach

The conventional approach to object recognition involves finding a correspondence between features in the image and features in given models (e.g., template matching; see Ballard and Brown, 1982). One extracts prominent features from the image and tries to match them with features in a model. Given the coordinates of k features in the image and the coordinates of k features in a possibly corresponding model, we calculate hypothetical parameters of the viewpoint, or pose, between the image and the model. This can include rotation, translation and other parameters. Repeating the calculation for other k-tuples, the results will cluster into the correct value of the pose parameters and into the correct model, which thus identifies an object in the image with this model. Having a number n of k-tuples of features and m models, we have to check $O(mn)$ possible matches, i.e. a number proportional to mn.

Several problems arise in this process. (1) Projection. The object we observe as well as the models in the database are three-dimensional, but the projection in the image is two-dimensional. Thus the depth information is lost and cannot be matched. (2) Viewpoint dependency. An image of an object depends on the point of view, including translation and rotation. (3) Complexity of finding the correspondence. This is in fact a consequence of viewpoint dependency. The fact that the visible object differs from the model in several parameters makes it impossible to match features without trying many matches. The number $O(mn)$ of possible matches mentioned above can be very large. Although there is no need to check all possible k-tuples of features in the image, the number we do check is usually many thousands. When we have hundred of models, the number of trial matches becomes prohibitive. (4) Reliability. Images contain noise and errors that corrupt measurements of the feature coordinates.

A common way to solve the projection problem is by using multiple images (stereo, motion). This can provide the missing depth information. The problem here is that now we must find the correspondence between features in at least two images. The complexity here is $O(n^2)$, assuming we use n k-tuples in both images. This is in addition to the problem we had earlier, of finding the correspondence between image features and model features.

Our approach will handle all the problems mentioned above. It will reduce the complexity by orders of magnitude, while solving the viewpoint dependency and projection problems.

At the heart of our approach to dealing with the problems described above are discoveries about the mathematical relations between a 3D object and its 2D projections. These are described in [Weiss96a,Weiss96b].

The key to solving the complexity problem is to make all our subsequent treatment viewpoint-invariant. A viewpoint change is generally represented by a projective transformation. However, when an object is distant from the camera (relative to the focal length), a projective transformation is very well approximated by an affine one. This condition is usually met in practice. An affine transformation can include translation, rotation, scaling, skewing and reflection. The latter includes switching from one half of a skew-symmetric image of an object to the other half. Instead of the k-tuples of features, we will use quantities, calculated from the coordinates of the features, that are invariant to affine transformation. These invariants will be used for matching both in the image-to-model case and the image-to-image (stereo, motion) case. In addition to k-tuples of pointlike features we will deal with conics and use their invariants.

Invariants have been successfully used before for planar objects (e.g. [Weiss88; Mundy92; Weiss93a; Weiss93b; Rivlin95]). There are no true invariants of the projection from 3D to 2D, but one can find invariant constraints that serve a similar purpose ([Jacobs92; Stiller94; Weiss96a; Weiss96b]).

We now describe the invariant mathematical relation between a 3D point set and its 2D projection. Details are in [Weiss96a]. We have a 5-tuple of points, with 3D coordinates $\mathbf{X}_i$, $i = 1 \ldots 5$. They are projected onto $\mathbf{x}_i$ in the image.

Since determinants are (relative) invariants of an affine transformation, we look at the determinants formed by these points in both 3D and 2D. Any four of the five points in 3D, expressed in four homogeneous coordinates, define a determinant M_i. We give the determinant the same index as the fifth point that was left out. For example,

$$M_1 = |\mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5|$$

Similarly, in the 2D projection, any three of the five points define a determinant m_{ij}, with indices equal to those of the points that were left out, e.g.

$$m_{12} = |\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5|$$

It can be shown by simple algebraic methods that the 3D and 2D invariants, M_i and m_{ij}, are related by the equations

$$M_5 m_{12} + M_1 m_{25} - M_2 m_{15} = 0$$
$$M_5 m_{13} + M_1 m_{35} - M_3 m_{15} = 0$$

These relations are obviously invariant to any affine transformation in both 3D and 2D. A 3D transformation will merely multiply all the M_i by the same constant factor, which drops out of the equations. A 2D affine transformation multiplies all the m_{ij} by the same constant factor, which again drops out.

The 2D invariants m_{ij} are calculated from measurements on the image. Thus the above relations are linear equations for the unknown 3D (relative) invariants M_i. There are three independent unknowns, namely the absolute invariants:

$$\frac{M_1}{M_5}, \quad \frac{M_2}{M_5}, \quad \frac{M_3}{M_5}. \qquad (1)$$

Since we only have two equations, we can determine the unknowns only up to one linear parameter. This remaining unknown represents the missing depth information.

3 Implementation Plan

Based on the above, we can build a 3D invariant space in which recognition will take place. Each 3D 5-tuple will be represented as a point in this space, with three invariant coordinates being the quantities shown in eq. (1). A model will be represented by a set of points in this space. (There will also be some way of identifying these points as part of the same model.) The problem is now how to match the image to the model. We will distinguish several cases.

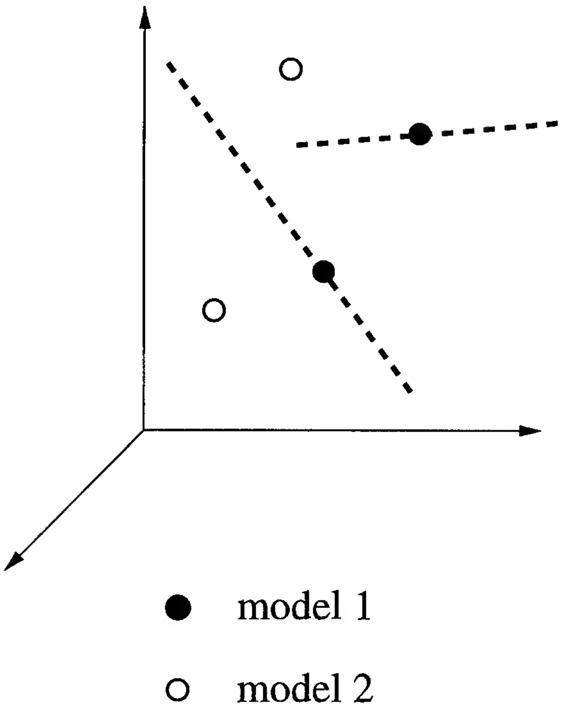


Figure 1: Single view recognition

3.1 Single image

This case is the simplest but least reliable. Here we extract 5-tuples of features from the image and calculate their 2D invariants m_{ij}. Substituting them in the above equations, we obtain two equations relating the three 3D absolute invariants of eq. (1). Geometrically, we obtain a line in our 3D invariant space (Fig. 1). If a 5-tuple in the 2D image is a projection of some 5-tuple in 3D, then the line obtained from this 2D 5-tuple will pass through the point in invariant space representing the 3D 5-tuple. To recognize objects we thus look for instances in which lines obtained from the image pass through points representing models in the invariant 3D space.

While this might work for a small number of models, we suspect that with many models and objects, producing many inexact lines and points, there will be many lines that pass through or near points that do not belong to them in reality. We need to find a way to eliminate the extra parameter along each line and reduce it to a point.

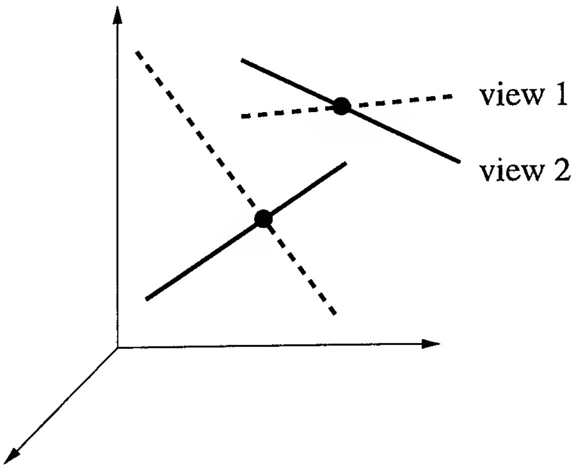


Figure 2: Two view matching

3.2 Multiple images

The first stage here is similar to the above, i.e. extract a 5-tuple of features from one of the images and transform it into a line in the 3D invariant space.

Next, we look at a different image, in which objects are seen from a different viewpoint. Again we draw a line in the same 3D space of invariants as before. If this line meets the first line in 3D then the two 5-tuples have the same 3D invariants (Fig. 2). These are the coordinates of the intersection point. This means that the two 5-tuples are affine equivalent. This in turn indicates that we may have two different views of the same 5-tuple.

With n 5-tuples, the total number of lines in the invariant space is $O(n)$. We can find line intersections in a way similar to that used in Hough space, namely divide the space into bins and see if a certain bin has more than one line going through it. We do not need to check all bins; we only need to go along the known lines. A hierarchical scale-space approach can be used to make the process more efficient. The exact technique and its cost need to be investigated, but it should be about $O(n)$. Therefore our total complexity is $O(n)$ rather than $O(n^2)$ as in previous methods.

Based on the above discussion, our recognition algorithm for multiple views involves the following steps:

i) Feature extraction. Find candidate 5-tuples that are potentially affine equivalent. These can be sets of points in some window placed in a similar location in both images, and they can overlap. Various clues can be used to prune unpromising sets, e.g. non-matching links that are visible between the points, or parallelism.

ii) Invariant description. For each 5-tuple that remains from (1), and for each view, calculate the two equations for the three 3D invariants, namely plot a line in a 3D invariant space. Find all lines that meet in 3D. Each intersection represents the 3D affine invariants of two affine equivalent 5-tuples.

iii) Recognition. We now have invariant 3D point sets representing various visible objects. Similarly, the models in the database can be represented by point sets in the same invariant 3D space. We will use several 5-tuples in each model, obtaining several points representing the model in the invariant space. Identification is now straightforward. If an object's point set falls on a model's point set in the invariant space, than the object is identified with the model. No search is needed to find the right model, because the invariant point sets of the models can be indexed according to their coordinates. No search is needed for the viewpoint or pose either because of the invariance. As many points as practical will be used to increase reliability.

iv) Verification. This step is independent of the invariants method. It overcomes any errors that we may have in calculating the invariants. Using the 2D coordinates of the images, and the correspondence found in (iii), we calculate the 3D coordinates of the features. Now we can find the 3D transformation (pose) that produces the best fit between the 3D object and the model identified in (iii), using least squares fitting. The identification is rejected if the fitting error is too big. We may try to fit several models to each object to find the best fit.

3.3 Single image, symmetric models

The problem encountered earlier in the single-image case was the unknown parameter along the line, resulting from the missing depth information. The only way to compensate for this in a single image is to use a model-based approach, i.e. make a modeling assumption about the 3D object.

One modeling assumption that we will use is symmetry. Most man-made objects are symmetric, e.g. vehicles, tanks, airplanes and buildings. Symmetry does not need to be bilateral; the Pentagon building is an example of a five-fold symmetry. Symmetry is also found in human and animal bodies. The (Euclidean) symmetry is observed explicitly only in a frontal view. In any other view the symmetry is observed as a skew-symmetry (i.e. it is an affine or projective symmetry). Many researchers have used skew-symmetry for recognition, but with serious limitations. They usually assume that the skew-symmetry itself is known, i.e. we know which feature in one half of the object corresponds to which feature in the symmetric half. In other words, they assume that the correspondence problem has already been solved. Here we will make no such assumption but will detect the skew-symmetric objects in an image.

The two halves (or three thirds, etc.) of a skew-symmetric object are affine equivalent. Therefore we can apply the algorithm described above which was designed to find affine equivalent 5-tuples. Having found matching 5-tuples, we have to verify that they are halves of the same object. The lines connecting corresponding points in a symmetric object are parallel in 3D, therefore they will be parallel in an affine projection, and this is easy to check.

The verification step (iv) is easier because of the symmetry assumption. The skew-symmetric object that we have found can be rectified, using an affine transformation, to obtain a standard view in which the object is (non-skew) symmetric. It can then be matched directly with a database of symmetric models.

4 Non-pointlike Features

The above discussion deals with point-like features. Other types of features can be useful, e.g. lines, conics and higher order curves. Conics arise in many cases from circular parts of objects, projected as ellipses. There are several advantages to using them. (1) There are far fewer of them than there are points. Therefore the complexity problem is greatly alleviated. (2) They are more reliable than points, because each conic is made up of many points whose measurement errors tend to average out.

Earlier work used invariants of a pair of conics for recognition. A serious limitation was that the two conics had to be coplanar. In [Weiss96b] we remove this restriction and deal with arbitrary conics in 3D. Given two uncalibrated images of a pair of conics in an arbitrary 3D configuration, we find the correspondence properties (the epipolar geometry) of the two images, and the parameters of the two conics, up to a 3D affine transformation.

Based on this we can devise a recognition algorithm as follows. Given two images, we pick a pair of conics in each. Using our method we find the epipolar geometry parameters of the images. We repeat the process with other conic pairs. With several such pairs, the parameters of the epipolar geometry will cluster around the correct values. The complexity here is low because of the small number of conics. From this we can find the 3D parameters of all conics (up to a global affine transformation, i.e. we can find affine invariant versions of the visible objects). These will be compared to the appropriate invariant parameters of conics from the models. These latter invariants will be indexed in a database of models.

This algorithm can be combined with the point-like feature based algorithm. Finding the epipolar geometry makes it easier to find the correspondence between point-like features in a stereo pair, further reducing the complexity involved in the algorithm described earlier.

5 Reliability

Reliability lies in numbers. Any system that has to deal with noisy inaccurate elements needs large numbers of them to obtain a reliable output. There is no shortage of information elements even in a single image; the problem is how to put them together. We can increase the numbers, and thus the reliability of our approach, in several ways:

1) Point sets will be extracted not as isolated features but as intersections of lines. A line contains many points and is more reliable than a single point.

2) Many point sets will be used for a single object. A match of one or two point sets may not be very reliable, but several matches involving the same object can yield a positive identification.

3) A verification stage, independent of the invariance method, is added as described earlier in step (iv). Several candidate models will be fitted to each visible object and the best fitting model will be chosen.

4) Curves such as conics will be used in addition to points, as described earlier.

5) Parts of objects will be recognized independently, e.g. the wings of an aircraft, to provide verification.

All these methods will be tested as described in the next section.

6 Evaluation Plan

Our main concern in evaluating our system's performance is reliability. We have proposed several ways to increase reliability, including the use of many features per object, and the addition of a verification stage which is independent of invariants. The effect of these and other factors on reliability will be tested.

The reliability of the system in terms of detection rate and false alarms depends on many variables. Some of these variables are inherent in the data, while others are specific to the system. The variables of the images are not under our control since we will use government-supplied real imagery . These variables include noise and errors in the images, the apparent object sizes, the angles of view, and the disparities between stereo images. The system variables are under our control. These include the number of models in the database, the size of the windows in which we are looking for the objects, the number of features used per model, and to some extent the choice of features.

The dependence of the detection/false alarm rate on these variables is understood qualitatively. The rate is higher if we have lower noise, bigger apparent size of the objects, fewer models in the database, etc. The evaluation plan will aim at quantifying these dependencies. It will consist of the following:

1) Conduct experiments to determine the quantitative dependence of the detection/false alarm rate on variables that are under our control. To this end we will keep the detection rate constant (e.g. 100% or close to it). We will look at the same data, and we will vary the system variables until we achieve the desired detection rate for those data. In this way we will optimize the system for the particular dataset that we have chosen.

2) Using the optimized system parameters, we will conduct experiments on other sets of data, with different characteristic variables, and determine the detection/false alarm rates. We will determine quantitatively how various variables which are not under our control affect the performance.

The output from these experiments will be interpolated as a surface in a multidimensional space. The surface "height" will represent the detection rate, while each abscissa will represent one of the variables involved.

Since both the data and the software will be available to the IU community, similar evaluations can be performed by other groups using our system. To achieve battlefield utility, very high standards of performance will be needed, such as detecting hundreds of objects of rather small apparent sizes in a battlefield filled with smoke and dust.

References

[Bal82] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, 1982.

[Jacobs92] D. Jacobs. Space Efficient 3D Model Indexing. *Proc. CVPR*, 439-444, 1992.

[Mundy92] J.L. Mundy and A. Zisserman, Eds: *Geometric Invariance in Machine Vision.* MIT Press, Cambridge, MA, 1992.

[Rivlin95] E. Rivlin and I. Weiss. Local Invariants for Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**:226–238, 1995.

[Stiller94] P.F. Stiller, C.A. Asmuth, and C.S. Wan. Invariant Indexing and Single View Recognition. *Proc. DARPA Image Understanding Workshop*, 1423–1428, 1994.

[Weiss88] I. Weiss. Geometric Invariants of Shapes. *Proc. DARPA Image Understanding Workshop*, 1125–1134, 1988.

[Weiss93a] I. Weiss. Geometrical Invariants and Object Recognition. *International Journal of Computer Vision*, **10**:207–231, 1993.

[Weiss93b] I. Weiss. Noise Resistant Invariants of Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**:943–948, 1993.

[Weiss96a] I. Weiss. Model-Based Recognition of 3D Curves from One View. *Proc. DARPA Image Understanding Workshop*, 1251–1256, 1996.

[Weiss96b] I. Weiss. 3D Curve Reconstruction from Uncalibrated Cameras. University of Maryland CS-TR-3605, 1996. Presented at ICPR 96.

Wavelet-based Target Hashing for Automatic Target Recognition

Robert Hummel and Davi Geiger[*]
Courant Institute, NYU
251 Mercer street, New York, NY 10012
E-MAIL: hummel@cs.nyu.edu and E-MAIL: geiger@cs.nyu.edu

Abstract

The New York University (NYU) project on "Wavelet-based Target Hashing for Atutomatic Target Recognition (ATR)" has focused on the use of geometric hashing, and related object recognition strategies, for the detection and localization of targets in digital imagery, particularly in Forward Looking Infrared Radar (FLIR) and in Sinthetic Aperture Radar (SAR) imagery. There are three main aspects to this work:

1. The development of stable, robust, feature extraction methods that apply to specific image modalities, and enable discrimination based on feature vectors describing idealized "interesting" constructs in the sensor data;

2. Theories for matching extracted features to model features, whether the models are predicted or observed from training data, generally assuming that there exists a one-to-one association between a subset of the observed features with a subset of the model features;

3. Studies of the discriminability of targets in actual applications by making use of geometric hashing software for the recognition of models in observed imagery.

1 Introduction

In addition to the Automatic Target Detection and Recognition (ATD/R) University Research Initiative funded by DARPA for the development of these technologies, there are a number of related research projects at NYU that also feed into these themes. In particular, NYU is a prime contractor in the Moving and Stationary Target Acquisition and Recognition (MSTAR) program, and has been active in the development of a Match Module for the model-based SAR ATR system, and more recently has been addressing the problem of global, intelligent search by the model-based subsystem of the MSTAR algorithm suite. We briefly summarize the relevance of the MSTAR work below, noting how the ATD/R work has benefitted the MSTAR algorithm development, and noting how the MSTAR work has suggested problems and studies conducted in the ATD/R project.

In addition, we have been working with a Florida company called I-Math Associates, led by Alexander Akerman III, to deliver software for geometric hashing for ATR applications. We have structured this software in such a way that it can easily be modified for application to MSTAR, and has also been used in the ATD/R project.

[*]Part of this work has been supported by the Air Force (AFOSR) through the Grants F49620-96-1-0159 and GC123913 NGD. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the United States Government, or NYU.

Finally, we have been conducting a project, also supported by the Air Foce (AFOSR), focusing on robust recognition including articulated object contours. While this work has not been applied to articulated target recognition using SAR or FLIR targets (mainly due to the lack of data of articulated vehicles), we expect this to change in the near future.

The full body of work can be reviewed by accessing `http://simulation.modsim.nyu.edu/` (look under "Projects." This web page can also be located from `http://cs.nyu.edu/` under "Research Projects" and then under "Computer Vision Group."

In the following, we highlight four topics. We begin with the disciminability of targers mentioned in (3), as listed above. We then study matching, and discuss a use of the bipartite matching algorithm for generalized feature-to-feature matching. This work forms part of topic (2). We move to topic (1), considering feature extraction, and discuss in some detail the corner and multi-junctions detector that has been developed at NYU. The key point of this detector is that it does not depend on a prior separate edge extraction process and has been extensively tested. We finally discuss the scheme to recognizing articulated objects contours.

2 Discriminability of Targets from Features

Overall, the NYU projects in ATD/R emphasize object classification and identification, and presuppose a detection process. Accordingly, the algorithms form part of a body of work in "Recognition Theory" that attempts to match patterns of features to model pattern of features. The emphasis is on a sufficient degree of recognition to be able to distinguish, for example, a T72 from an M60 in FLIR imagery. Of course, we depend on a sufficient resolution so as to extract enough pixels on each target. Ultimately, a sufficient number of features are required so as to disambiguate the observed object from among the collection of models, and from the model for clutter and nonspecific targets. Some separate theoretical analyses suggest that this is possible if there are something like 30 or 35 individual, independent "features" on each target, where a feature (in this case) means a single coordinate (of a possibly higher-dimensional feature), represented by a real number. This ballpark estimate makes many assumptions, and will vary according to the background complexity and the specificity of the features on the target. However, for the kind of target versus background discrimination that is required using high-resolution imagery so as to keep false alarm rates to tolerable levels (something less than one false alarm per square kilometer of SAR imagery, for example), these estimates provide some guidance. Of course, since 35 independent features might come from a dozen locations in the image, each with an attribute measure, we conclude that recognition should be possible given perhaps a thousand pixels on the target. Fewer pixels on target might suffice when there are a dozen significant location features (with attributes) that can be extracted, but sufficient resolution is the key to being able to extract a sufficient number of features.

3 Bipartite Matching

The bipartite matching problem can be described as the problem of determining a minimum-cost one-to-one assignment of nodes from one collection to another collection, where the total cost is the sum of nonnegative individual constituent costs. In the field of computer vision, the object recognition problem can often be formulated as a problem of matching a collection of model features to extracted features, where the model features are suitably transformed into the image domain and where noise and obscuration can lead to nomatch conditions. Under the assumptions that the features are conditionally independent, it is possible to formulate a Bayesian match metric as a sum of individual terms based on the correspondences between model and image features. This model provides an explicit representation for noise and obscuration, manifested as unmatched extracted features and unmatched model features. The optimal assignment problem can be seen as a matching problem in bipartite graphs, and thus admits an efficient solution.

Our Model

In model-based computer vision, predefined models are hypothesized to be present in the scene and evidence is sought in the sensed image for supporting or rejecting each hypothesis. Search strategies are used to guide the hypothesis evidence accrual process, which continues until either a decision is made about the identity of the particular model(s) present and the geometric transformation that best superimpopses the model features into the image, or an empty decision is advocated, implying that the statistical evidence is not strong enough to support any of the current model hypotheses.

The object recognition problem then consists of finding an instance of (a subset of) the model embedded in the image subject to invariance under some group of transformations. We accomplish this recognition by matching model features to extracted features, and comparing the corresponding feature values. We are concerned with the efficient evaluation of a match metric in order to compare a hypothesized model with an observed scene. There are two aspects to this comparison: (1) Finding correspondences between extracted and model features, after the model has been suitably transformed to optimize the ultimate score, and (2) computing the score based on those associations by means of a Bayesian metric for the distance between the sets of features.

The Bayesian match score can be expressed as

$$\log\left(\frac{\mathrm{L}(\,\mathbf{X},\Gamma\mid T_k\,,\Theta_0)}{\mathrm{L}(\,\mathbf{X}\mid T_0\,)}\right) =$$

$$\sum_{\Gamma}\Big(\log(p_i)+\log f_i(\mathbf{x}_i-\mathbf{y}_i)-\log\rho_0(\mathbf{x}_i)\Big)\;+ \tag{1}$$

$$\sum_{\mathbf{Y}^u}\log(1-p_i)+\sum_{\mathbf{X}^u}\Big(\log\rho_i(\mathbf{x}_i)-\log\rho_0(\mathbf{x}_i)\Big).$$

which can be written as a sum of weights for a given assignment of model features to extracted features as follows. Let

$$\begin{aligned} c_{ij} &= \log(p_i)+\log f_i(x_j-y_i)-\log\rho_0(x_j) \\ a_i &= \log(1-p_i) \qquad\qquad\qquad\qquad (2) \\ b_j &= \log\rho_j(x_j)-\log\rho_0(x_j) \end{aligned}$$

Then consider z_{ij} to be a permutation matrix that will assign feature $\mathbf{y}_i$ to feature $\mathbf{x}_j$ by setting $z_{ij} = 1$. Likewise, we will set variables $y_i = 1$ when $\mathbf{y}_i$ is unmatched, and $x_j = 1$ when $\mathbf{x}_j$ is unmatched; all other variables z_{ij}, y_i and x_j are set to zero. Then using the definitions above, and in view of equation (1) the problem of chosing correspondences so as to maximize the resulting score becomes

$$\text{maximize} \qquad \sum_i\sum_j c_{ij}z_{ij}+\sum_i a_iy_i+\sum_j b_jx_j \tag{3}$$

subject to the following constraints on the variables:

$$\begin{aligned} \sum_j z_{ij}+y_i &= 1 \qquad \text{for} \quad i=1,2,\ldots,m (4) \\ \sum_i z_{ij}+x_j &= 1 \qquad \text{for} \quad j=1,2,\ldots,s \,(5) \\ z_{ij},\,x_i,\,y_j &\geq 0 \qquad \text{for all} \quad i,j. \qquad\quad (6) \end{aligned}$$

The constraints represent the fact that for each predicted feature, either there is exactly one match to an extracted feature, or the feature is unmatched, but not both. Likewise, for each extracted feature, either there is a match to a corresponding predicted feature, or the extracted feature is unmatched, but not both.

Thus the original problem can be seen as bipartite graph matching, where the m predicted features are supplemented with s extra nodes, one for each extracted feature to serve as potential no-match nodes, and the s extracted features are supplemented with m no-match nodes, one for each predicted feature (Figure).

As a result of these observations, we see that the problem of finding an association between predicted model features and extracted scene features in such a way as to maximize a score that

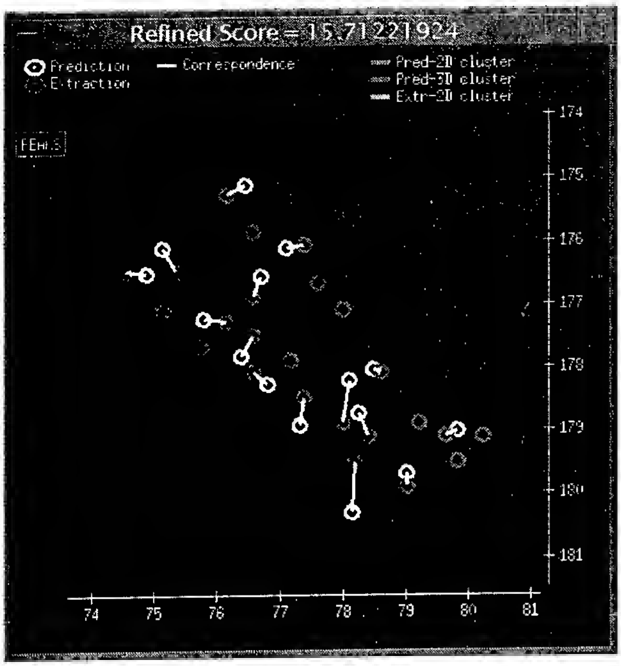


Figure 1: Peak correspondences of a military truck.

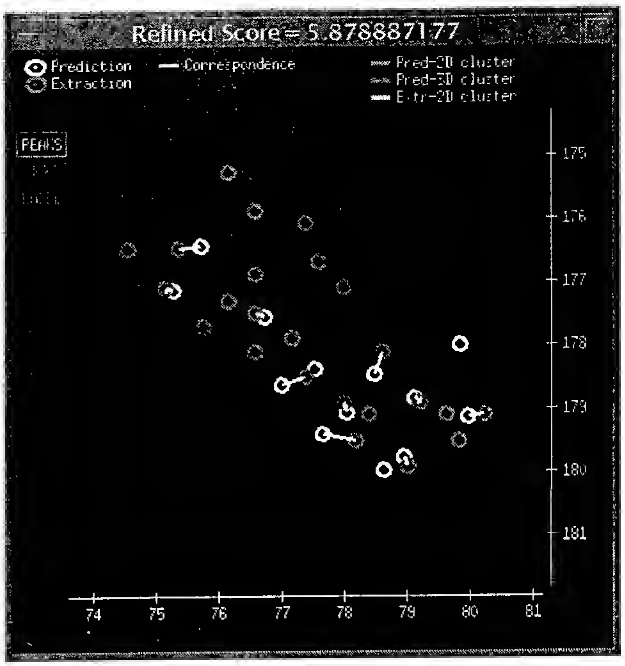


Figure 2: Peak correspondences between a predicted truck model and a SAR image of a tank.

accounts for penalties for unmatched predicts and unmatched extracts can be converted into a bipartite matching problem that can be easily solved using the Hungarian algorithm. Thus a combinatorial number of possible assignments can be examined in polynomial time, and an optimal assignment is obtained in terms of the match metric based on a Bayesian scoring. The essential ingredient permitting this optimal association is the structure of the match metric as a sum of nonnegative weights, and this formulation in turn depends on using log-likelihoods and conditional independence assumptions.

4 Junction Detection

Corners, T-, Y-, X-junctions give depth cues which is a critical aspect of image understanding tasks such as object recognition: junctions form an important class of features invaluable in most vision systems.

The three main issues in a junction (or most feature) detector are: (1) Scale; (2) Location; (3) Junction (feature) parameters. The junction parameters are: The radius or size; The number of wedges (lines, corners, 3-junctions such as T or Y, or, 4-junction such as X-junction, etcetera); The angles of each wedge; The intensity in each of the wedges. Most junction detectors, reported in literature, do not address all these issues in toto: very often ad hoc methods are used to address the scale or location issue. Our main contribution is a modeling of the junction (using the minimum description length principle), which is complex enough to handle all the three issues and simple enough to admit an effective dynamic programming solution.

Similar approach can be used to model other features like thick edges, blobs and end-points.

We have tested this detector against a large variety of imagery and tested the parameters against different levels of noise to conclude that we have a very robust junction detector. We have not yet tested the algorithm within a recognition system, which we plan to do this coming year.

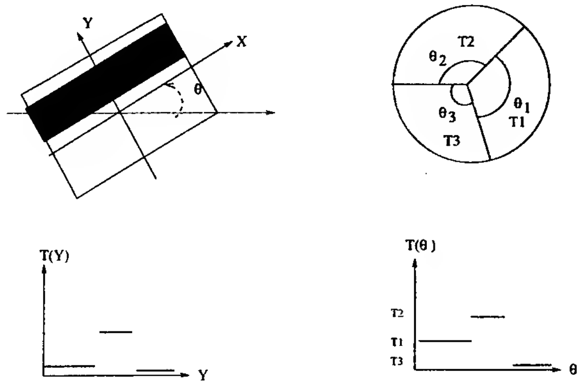


Figure 3: Piecewise constant features: a bar detector on the left column and a junction detector on the right column.

The Junction Model

We model a junction as a region of an image where the values are piecewise constant in wedge-shaped regions emanating radially from a central point, covering a small disk centered at the point and omitting a (much) smaller disc centered at this point (see figure 3). The parameters of a junction consist of (i) the radius of the junction-disk, (ii) the center location, (iii) the number of radial line boundaries, (iv) the angular direction of each such boundary, and (v) the intensity within each wedge. The radius of the disk addresses the "scale" issue, and the location of the center is a kind of "interest operator" that determines the position where the feature is located in a region, possibly pre-defined.

We can formulate the junction detection problem as one of finding the parameter values that yield a junction that best approximates the local data using minimum description, and declaring local minima of the error as junctions. The best-fit parameter values provide attributes of the detected junction.

Let T denote the piecewise constant function/template. It has N angles and N intensities if N is the number of constant pieces. Further, let I denote the input signal.

Define the energy function, at a point (i, j) on the image as follows

$$E = \mathcal{D} + \lambda \mathcal{G}\,,$$

where $\lambda \geq 0$.

The first term, $\mathcal{D}$, is a measure of the distance of the fitted function from the *data* using the L^2 norm:

$$\mathcal{D} = \int_0^\infty \int_0^{2\pi} [I(r,\theta) - T(\theta)]^2 g(r) r dr d\theta, \quad (7)$$

where $g(r)$ is an appropriate modulating function that goes to zero for large r, thus defining the template size.

The second term, $\mathcal{G}$, is a measure of the distance of the *gradient* using the L^2 norm.

$$\mathcal{G} = \int_0^\infty \int_0^{2\pi} |\nabla[I(r,\theta) - T(\theta)]|^2 g^*(r) r dr d\theta, \quad (8)$$

where $g^*(r)$ is an appropriate modulating function, not necessarily the same as $g(r)$.

Note that,

$$\nabla I = \frac{\partial I(r,\theta)}{\partial r}\mathbf{e_r} + \frac{1}{r}\frac{\partial I(r,\theta)}{\partial \theta}\mathbf{e_\theta}$$

$$\nabla T(\theta) = \frac{1}{r}\frac{\partial T(\theta)}{\partial \theta}\mathbf{e_\theta}$$

where $\mathbf{e_r}$ and $\mathbf{e_\theta}$ are the orthonormal vectors in the r and θ direction respectively, evaluated at (r, θ). So we obtain,

$$\mathcal{G} = \mathcal{A} + \mathcal{R},$$

where,

$$\mathcal{A} = \int_0^\infty \int_0^{2\pi} \frac{1}{r^2}\left(\frac{\partial I}{\partial \theta} - \frac{\partial T}{\partial \theta}\right)^2 g^*(r) r dr d\theta, \quad (9)$$

$$\mathcal{R} = \int_0^\infty \int_0^{2\pi} \left(\frac{\partial I}{\partial r}\right)^2 g^*(r) r dr d\theta. \quad (10)$$

We have considered

$$g(r) = \begin{cases} 0 & r < R_0 \\ \frac{1}{r} & R_0 \leq r \leq R_1 \\ 0 & r > R_1 \end{cases},$$

and $g^*(r) = r^2 g(r)$. Our rational to introduce a "hole" of size R_0 is that (i) It works better than without; (ii) Optical effects are removed. At the very center part of the junction we expect to find a blurred signal.

Selecting the Scale

A user-defined threshold bounds $\lambda\mathcal{R}$: this defines R_1. R_0 is often a user-defined fraction of R_1: this allows a small hole in the center. A significant observation, from the series of experiments, has been the use of non-zero R_0.

To select a best location in an image region: $\lambda\mathcal{R}$, (with not necessarily the same R_1), is evaluated at the points in the region. The one with the minimum value defines the location. We illustrate this in Figure 5.

Estimating wedge, angles and intensities

Let us assume the number of wedges is fixed. We propose a dynamic programming formulation. Let A_1, A_2, $\ldots$, A_K denote the range of admissible intensities, θ_1, θ_2, $\ldots$, θ_K, the discretized angles. Then, dynamic programming can minimize $\tilde{E} = \mathcal{D} + \lambda\mathcal{A}$ over the set $\{(A_i, \theta_i), i = 1, \ldots, K\}$ (for more details see [17]).

Estimating the number of wedges: The (optimal) number of wedges, N, is computed by thresholding the relative error, r^n,

$$r^n = \frac{\overline{E}^{n+1}}{\overline{E}^{n}}.$$

Although, in principle, we are looking for the minimum r^n, in practice we terminate the computation when r^n drops below a pre-defined threshold, as n is increased. Note that as the number of parameters increase, $\overline{E}^n$ decreases, i.e., $r^n < 1$. A variational form that may justify this approach is to use, as suggested in the book of Morel and Solimini [15], a split and merge algorithm for the energy $\tilde{E} = log(r^n)$.

To summarize, the following steps are involved in detecting junctions on a large region of an image:

- Compute $\mathcal{R}$, the measure of radial variation, and R_1, the size of the template at every point.
- Filter the locations using a threshold on $\mathcal{R}$.
- In a neighborhood of a filtered location, pick the one with minimum $\mathcal{R}$, and remove all other locations within a radius of R_1 of this. Repeat this for all the filtered locations.
- Compute the junction parameter for all the filtered locations.

4.1 Results of Experiments

We carried out a series of experiments on synthetically generated images (see in Figures 4) to test the stability of the algorithm in the presence of noise in the image, and to further understand the role of smoothing. In these experiments, we suppressed the use of threshold values using a fixed R_1 and fixed $N = 3$, i.e., forcing the algorithm to pick up the otpimal 3-corners. We then carried the experiments on real images to test the general performance of the algorithm.

Stability of the Algorithm: Figure 4 shows the result of one such experiment where we look for 3-corners in the center region of the image. It shows that when the image edges are sharp ($\sigma = 0 \ldots 3$ in the figure), the error in the angles and the intensities are slightly higher. The errors are least in the range of $\sigma = 6 \ldots 20$, and increase further on.

Note that the (Manhattan) distance of the location of the 3-corner is less than 6 pixels; the error in the angles is bounded by 30° at the very worst and the intensity differs by 40 units in the worst situation. It also shows that at the best situation the angle differs, on an average, by 15° from the true answer.

When the images used are the smoothed version of the ones shown in Figure 4, the reduction in error around the sharp images or low values of σ.

It is worth to note that even for sharp junctions (without noise added) the algorithm performs better when smoothing is introduced. This is because of the natural discretization of the image and the the numerical scheme to integrate $I(r, \theta)$ over r. This bias is clearly reduced when homogeneous smoothing is introduced prior to the numerical integration.

Figure 6 show the results of the detector on regions of images. After the filtering it computes the parameters for about four junctions, in a minute, on a Sun Sparc Station.

5 Articulated Object Recognition

Our approach is a Bayesian integration of *shape similarity* and *snakes*, and naturally combines top-down and bottom-up algorithms. The bottom-up method extracts edges, then constructs snakes (or contours) by grouping edge elements and feeds the shape analysis. The top-down one uses shape analysis, by comparing the object model with the extracted snakes, to guide/prune the search for other snakes. The optimizations are based on Dijkstra algorithm and further pruning of this algorithm is obtained by working on object parts separately. Our approach is general enough to handle three dimensional objects, but our focus here is on two dimensional contours.

The Model

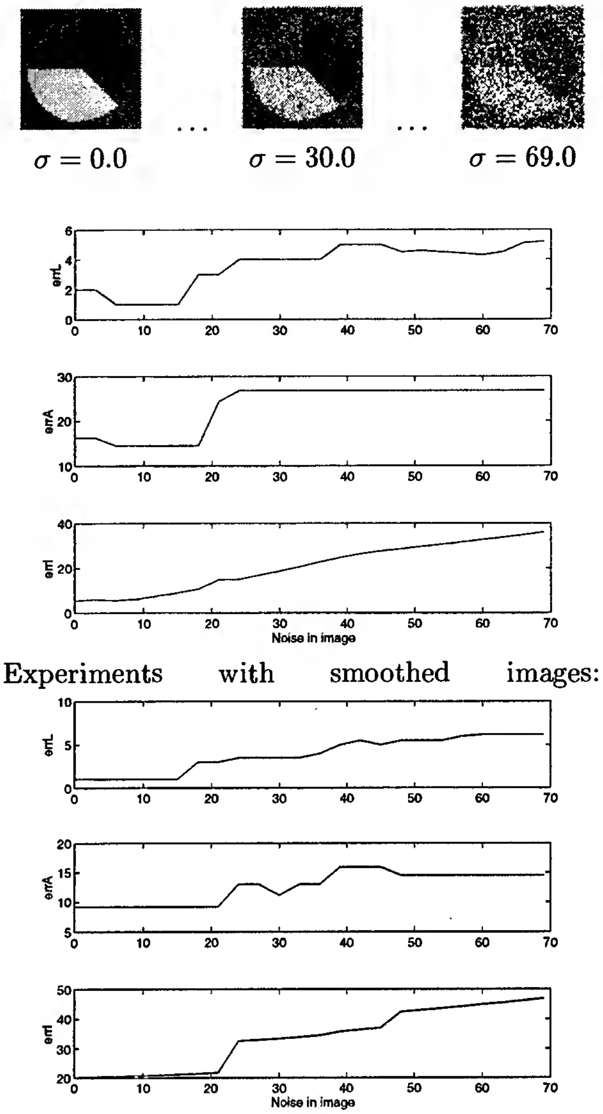
We are given an image I and first establish the difference and similarity between the processes of recognizing and of finding objects in images.

In order to recognize a contour model Γ^S in the image, we construct

$$
\begin{aligned}
&P(\Gamma^S|I) \\
&= \textstyle\sum_{(\Gamma^T,\{t(s)\})} P(\Gamma^S,\Gamma^T,\{t(s)\}\,|I) \quad (11)\\
&= \textstyle\sum_{(\Gamma^T,\{t(s)\})} P(\Gamma^S,\{t(s)\}\,|\Gamma^T,I)P(\Gamma^T|I)(12)
\end{aligned}
$$

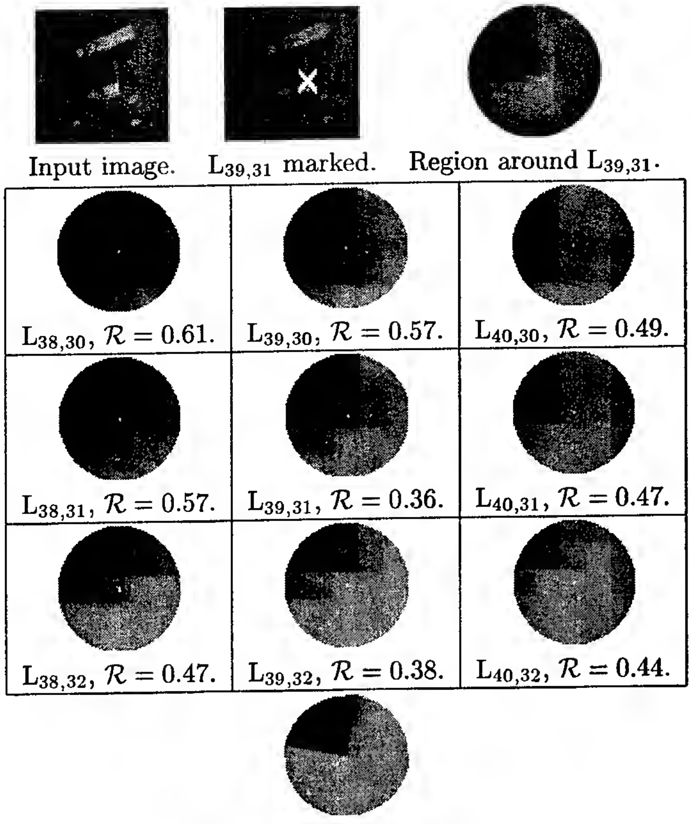
where the sum is over all possible image contours Γ^T and all possible correspondences, $\{t(s)\}$, between the image contour Γ^T, parametrized by t, and the model contour Γ^s, parametrized by s. Note that while we may not know $P(\Gamma^S|I)$ we can create simple generative models for contours to obtain $P(\Gamma^T|I)$.

The problem of finding contours in images can be thought as the one of finding the contour Γ^{T*}, in the image, that most contribute to the sum above, i.e. to find



Experiments with smoothed images:

Figure 4: Test of stability: 8-bit images with Gaussian noise. The first image in the top left has standard deviation $\sigma = 0$ (that is, no noise) and the angles of the 3-corner at the center, (x,y), are $(a_{1f},a_{2f},a_{3f}) = (90°,180°,315°)$, with intensities $(i_{1f},i_{2f},i_{3f}) = (120,200,40)$. Noise (σ) is varied from 0 to 69 to obtain 24 images, three of which are shown in the top row for illustration. The following errors are computed: location, $errL_\sigma = |x_\sigma - x| + |y_\sigma - y|$, intensities, $(errI_\sigma)^2 = \sum_{j=1}^{3}(i_{j\sigma} - i_{jf})^2/3$, angles, $(errA_\sigma)^2 = \sum_{j=1}^{3}(a_{j\sigma} - a_{jf})^2/3$ and plotted vs σ. These images are now Gaussian smoothed, with increasing factor as the image gets noisier. The first eight images are smoothed using a $\sigma_s = 2.0$, the next eight uses $\sigma_s = 3.0$ and the last set uses $\sigma_s = 4.0$. We plot the errors as in the previous one. As expected we are able to get rid of the high errors for the very sharp images (low values of σ).

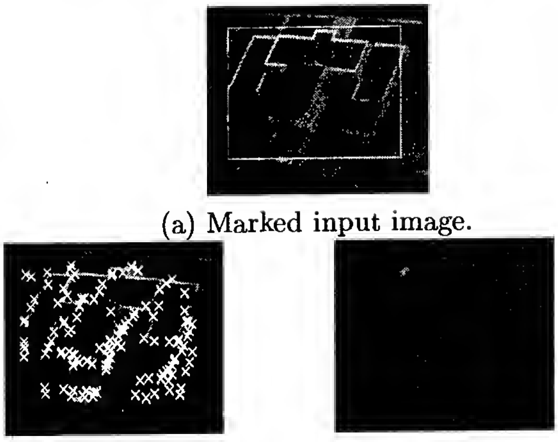


Input image. $L_{39,31}$ marked. Region around $L_{39,31}$.

$L_{38,30}$, $\mathcal{R} = 0.61$.	$L_{39,30}$, $\mathcal{R} = 0.57$.	$L_{40,30}$, $\mathcal{R} = 0.49$.
$L_{38,31}$, $\mathcal{R} = 0.57$.	$L_{39,31}$, $\mathcal{R} = 0.36$.	$L_{40,31}$, $\mathcal{R} = 0.47$.
$L_{38,32}$, $\mathcal{R} = 0.47$.	$L_{39,32}$, $\mathcal{R} = 0.38$.	$L_{40,32}$, $\mathcal{R} = 0.44$.

The junction at location $L_{39,31}$.

Figure 5: The use of $\mathcal{R}$, to locate the center of the junction. $L_{x,y}$ indicates the x and y coordinates on the image. Note that the location $L_{39,31}$ has the minimum $\mathcal{R}$ in the neighborhood. Incidentally, R_1, the size of the template is the same for all the nine locations.



(a) Marked input image.

(b) Junctions. (c) Junction templates of (b).

Figure 6: Image results. Low contrast junctions have not been filtered.

$$\Gamma^{T*} = arg \max_{\Gamma^T} \sum_{\{t(s)\}} P(\Gamma^S, \{t(s)\} \,|\Gamma^T, I) P(\Gamma^T|I) .$$

The focus of our work becomes to compute the optimal $(\Gamma^{T*}, \{t^*(s)\})$.

Key points and features

We introduce the possibility of detecting localized features in the image represented by a set $\{y_k \; : k = 1, ..., K\}$. They can represent corners, junctions, high curvature points, etc. Since our models are contour models, we use an ordered set of model features $[x_s \; : s = 1, ..., S]$ representing possibly high curvature points, or bending points.

The key points of a model contour are points that convey a lot/most of information about the contour (see Figure-7(a)). These points connected by straight lines can give a good caricature of the contour, even when this contour undergoes deformations/articulations. All the points of a contour figure that are allowed to bend (articulation points) are included. It is beyond the scope of this paper to derive/detect these points from an information-theoretic modeling. We will simply manually select them for each contour model.

Note that the key points of the model can, in a limit case, be all of the contour points. The distance between two points in the model, s and r are given by $d_S(s, r)$ and can be precomputed by following the contour. We then decompose $P(\Gamma^T|\Gamma, I)$ as

$$\begin{aligned}
&= \sum_{[x_s],\{y_k\}} P(\Gamma^T, [x_s], \{y_k\}|\Gamma, I) \\
&= \sum_{[x_s],\{y_k\}} P(\Gamma^T|[x_s], \{y_k\}, \Gamma, I)\; P([x_s], \{y_k\}|\Gamma, I) \\
&\approx \sum_{[x_s],\{y_k\}} P^{shape}(\Gamma^T|[x_s], \{y_k\}, \Gamma)\; P^{snake}(\Gamma^T|\{y_k\}, I) \\
&\qquad P^{features}(\{y_k\}|[x_s], \Gamma, I)\; P^{model}([x_s]|\Gamma) , \quad (13)
\end{aligned}$$

where in the approximation we did consider the model points, $[x_s]$, to be independent of the image I, i.e., $P^{model-points}([x_s]|\Gamma) \;=$

$P^{model-points}([x_s]|\Gamma, I)$. $P(\Gamma^T|[x_s], \{y_k\}, \Gamma, I)$ is decomposed into a shape model, $P^{shape}(\Gamma^T|[x_s], \{y_k\}, \Gamma)$ and a snake model $P^{snake}(\Gamma^T|\{y_k\}, I)$. We say the whole process is a deconstruction because we not only decompose the recognition into various terms, but later we reconstruct the final object in the image through this decomposition.

Ideally, the image features $\{y_k\}$ are originated from the model points $[x_s]$. A model of the geometric projection as well as illumination and reflectance is necessary. However, in our experiments we have neglected this issues and simply used the "SUSAN" feature detector [21] to generate a set of key features in the image. They are candidates for matching the model key features. These features usually capture corner points, junction points, and salient edges and we have found them useful for our experiments (see Figure-7(b)). Clearly, the total number of detected features affect the algorithm complexity and we do require a reasonable amount of them to be capable of detecting the target. Roughly speaking, the feature resolution should be fine enough to sample a similar distribution like the one found by those model key features.

Matching key points and features: In the above model we have not made explicit that the model points , $[x_s]$, have a correspondence in the feature set $\{y_k\}$. We can make so by introducing binary matching variables $\{M_{s,k} \ : s = 1, ..., S : k = 1, ..., K\}$, so that

- $M_{s,k} = \begin{cases} 1, & \text{when } y_k \text{ is matched to } x_s, \\ 0, & \text{otherwise.} \end{cases}$
- $\sum_{s=1}^{S} M_{s,k} = 0$ or 1. 0 occurs when y_k is not part of the model contour.
- $\sum_{k=1}^{K} M_{s,k} = 1$.

In this paper, we have not experimented with occlusions; otherwise we would need to consider the case $\sum_{k=1}^{k} M_{s,k} = 0$.

Top-down and Bottom-up

We can (see [8]) approximate (13) into $P(\Gamma^T|\Gamma, I) \approx$

$$\prod_{s,i,j=1}^{S,K,K} e^{-M_{s,i}M_{s-1,j}[E^{sn}_{i,j}((\Gamma^T)^{y_i}_{y_j}, I)+\beta\ E^{sh}_{s,i,j}((\Gamma^T)^{y_i}_{y_j}, (\Gamma)^{x_s}_{x_{s-1}})]}, \quad (14)$$

where

$$E^{sn}_{i,j}((\Gamma^T)^{y_i}_{y_j}, I) = \int_{(\Gamma^T)^{y_i}_{y_j}} \left[\frac{1}{|\nabla I|^2(t) + \epsilon} + \lambda_1 k^T(t) + \lambda_2\right] dt\,,$$

and

$$E^{sh}_{s,i,j} = \frac{[\Theta_S(s) - \Theta_T(i)]^p}{[|\Theta_S(s)| + |\Theta_T(i)|]^{p-1}} + \lambda \frac{|t'(s,i,j) - 1|^p}{(t'(s,i,j) + 1)^{p-1}}\,.$$

We now give an intuitive account, the rational, of the top-down & bottom-up processes. Following that we derive this rational from the optimization method, Dijkstra algorithm, to compute the solution.

The rational: From the top-down view of the problem, we need to select the optimal choice of M, i.e., the optimal matching correspondence between model key points and image key features. The algorithm must not search for all possible configurations of M to be efficient. The guidance of the top-down model is to decide which pairs, say $M_{s-1,j} = 1$ and $M_{s,i} = 1$, to consider, without searching for all possible ones.

The top-down module hypothesizes a pair of correspondences, say $M_{s-1,j} = 1$ and $M_{s,i} = 1$, and feeds the model contour Γ^s_{s-1} to the bottom-up module.

The bottom-up process test the hypothesis that (x_{s-1}, x_s) corresponds to (y_j, y_i), i.e., $M^*_{s-1,j} = 1$ and $M^*_{s,i} = 1$, taking the contour part Γ^s_{s-1} as input. There is freedom to select different contours $(\Gamma^T)^{y_i}_{y_j}$ connecting y_j to y_i. The bottom-up process returns the cost induced by the best contour choice, $(\Gamma^{T*})^{y_i}_{y_j}$. This cost takes into account the snake cost and the shape comparison cost with Γ^s_{s-1}.

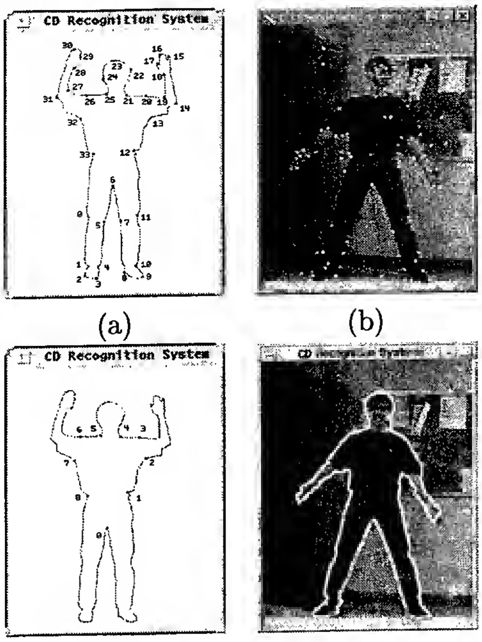
The top-down will then decide which other pair of correspondences to hypothesize for its search (see more details in [8]). As a motivation to read our article we show the results of the algorithm.

References

[1] M. Akgül, *The Linear Assignment Problem.* In: Combinatorial Optimization: New Frontiers in Theory and Practice. Springer-Verlag, 1992.

[2] D. Beymer, Massachusetts Institute of Technology Master's thesis, *Junctions: Their detection and use for grouping images*, 1989.

[3] V. Caselles, B. Coll, J. M. Morel, *A Kanizsa Programme*, Technical Report, Universitat de les Illes Balears, Spain, 1996.



(a) (b)

(c) (d)

Figure 7: (a) Original model with its key points. (b) Original image with key features. (c) the model contour. (d) corresponding shape boundary detections.

[4] R. Deriche, T. Blaszka, *Recovering and Characterizing Image Features Using An Efficient Model Based Approach.* In Proceedings of Computer Vision and Pattern Recognition, New York, 1993.

[5] W. Forstner, E. Gulch, *A Fast Operator for Detection and Precise Location of Distinct Points, Corners, and Centres of Circular Features,* Procc of Intercommssion Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland, 1987, pp 281-305.

[6] W. Freeman, E. Adelson, *Junction Detection and Classification,* Proc. ARVO 1991.

[7] D. Geiger, R. Hummel, A. Baldwin, L. Liu, L. Parida. Feature transform for ATR image decomposition, *Proc. SPIE Aero*, Florida, 1995.

[8] D. Geiger and T. Liu. Recognizing articulated objects with information theoretic methods, *International Conference on Automatic Face and Gesture Recognition* Killington, Vermont 1996.

[9] G. Giraudon and R. Deriche. *On corner and vertex detection.* In Proceedings of Computer Vision and Pattern Recognition, Hawaii, 1991.

[10] A.V. Goldberg and R. Kennedy, *An Efficient Cost Scaling Algorithm for the Assignment Problem.* Technical Report, Stanford University, 1993.

[11] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1988.

[12] A. Guzman, *Decomposition of a Visual Scene into Three-Dimensional Bodies*, Proc. AFIPS 1968 Fall Joint Computer Conference, 1968.

[13] M. F. Hueckel, *An operator which locates edges in digitized pictures*, J, Assoc. Compt. Mach. Vol 18, 1971.

[14] R. Hummel, *Feature Detection Using Basis Functions*, Computer Graphics and Image Processing, Vol 9, 1979.

[15] J. Morel, S. Solimini, Variational Methods in Image Segmentation, Birkhauser Boston, 1995.

[16] M. Nitzberg, D. Mumford, T. Shiota, *Filtering, Segmentation, and Depth,* Springer Verlag Berlin 1993.

[17] L. Parida, D. Geiger, B. Hummel, Junction Detection Using Piecewise Constant Functions *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Venice, 1997.

[18] A. Pentland, "Recognition by Parts," *Proceedings of the First International Conf. on Computer Vision,* pp. 612-620, 1987.

[19] I. Rigoutsos and R. Hummel, *A Bayesian approach to model matching with geometric hashing.* Computer Vision and Image Understanding, Vol **62**, Num 1, 1995.

[20] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Annals Statistics, vol. 11, pp, 416-431, 1983.

[21] S. Smith and J. Brady, "SUSAN - a New Approach to Low Level Image Processing," *Int. Journal of Computer Vision,* 1996, In publication.

Image Understanding Research For Battlefield Awareness at Johns Hopkins University

Lawrence B. Wolff *
Computer Vision Laboratory
Department of Computer Science
The Johns Hopkins University
Baltimore, Maryland 21218
wolff@cs.jhu.edu

Abstract

Innovations in Polarization Vision Technology made over the past several years have enabled the recent preliminary demonstration of unique capabilities for battlefield awareness provided by polarization sensing. Just underway is a comprehensive effort in polarization-based Automatic Target Recognition and Detection (ATR/D) encompassing both scientific and empirical aspects. Overviewed are the objectives, research issues, and, methods of performance evaluation for this new effort.

1 Motivation

The main concept motivating this effort is that passively sensed characteristics of polarization of reflected, scattered, and, thermally emitted light can be predictably related to key physical characteristics of man-made objects (e.g., military target vehicles, camouflage netting, etc...) and to physical characteristics of natural background and terrain. This is leading to the systematic development of Automatic Target Recognition and Detection (ATR/D) image understanding algorithms with unique capabilities that can be of large importance to battlefield awareness. Polarization-based image understanding provides unique advantages in areas that have been especially problematic using other existing technologies, including: i) ATR/D under heavy occlusion and heavy clutter, ii) detection of camouflage netting, and detection of presence of target vehicles under such netting, and, iii) distinguishing actual targets from false decoys. In addition, polarization cues provide important augmented physical features for target recognition and target pose determination. The following summarizes some key points:

- Polarization being physically orthogonal to intensity and color is invariant to *extrinsic appearance*, either modified by camouflage and/or degraded by clutter.
- Reflected and thermally emitted polarization is directly related to *intrinsic properties* of a target including material composition, surface roughness, and 3D shape.
- Polarization sensing has significant advantages over appearance based ATR/D methods not directly linked to physical characteristics of targets.

A major challenge to ATR/D image understanding is the unpredictable non-repeatibility of features due to signature variations of targets in different physical states, further confounded by coincidental appearance of cluttered scene elements and intended deceptive appearance of decoys. The use of polarization in this effort predictively links distinctive physical signatures to man-made objects dependent upon physical characteristics such as local orientation, local change in orientation (i.e., local "bending" or curvature), material roughness, and, material composition while being invariant to brightness and color appearance states. With validation from supporting reflectance models, reliable levels of repeatable algorithm performance can be achieved by predictively linking signatures to physical characertistics that distinguish man-made objects from natural background (including cluttering) for target detection, and that distinguish between man-made objects for target recognition and identification of decoys. Furthermore, polarization signatures can be used to solve backwards estimating some aspects of the state of a target vehicle, such as its relative pose. The innovation of polarization camera technology, developed by the principal investigator under previous DARPA efforts, now makes it possible to reliably acquire large data collections of empirical polarization imagery for algorithm development, test and performance evaluation, and direct comparison with predictive reflectance modeling.

Successful completion and demonstration of this effort could significantly impact the state-of-the-art in ATR/D with the development of a new sensory class of image understanding algorithms that transcend conventional methods of disguising target assets using camouflage ap-

*Past research funding that has contributed to this effort include DARPA grants DAAH04-94-G-0278, DAAH04-96-1-0216, DARPA contract F30602-92-C-0191, AFOSR grant F49620-93-1-0484, and NSF National Young Investigator Award IRI-9357757.

pearance methods such as with paint, netting, and, cluttering of background. Even further- countermeasures to defeat polarization-based ATR/D image understanding algorithms will require a serious rethinking by military contractors about the design and construction of camouflage netting and target decoys, what intrinsic material properties paints will have to possess that cover target vehicles, and, even possible reconsideration of the exterior geometric characteristics of military vehicles themselves. While the basic research aspects of this effort in terms of predictive polarization reflectance modeling are primarily intended for the development and validation of polarization-based ATR/D algorithms, they will in fact simultaneously provide significant insight into just what countermeasures are needed to defeat such algorithms.

2 Objectives

The highest level objective of this effort is to demonstrate a definitive proof of concept of polarization vision as a new innovative technology having significant importance for Automatic Target Recognition and Detection for battlefield awareness.

2.1 Scope

This is a comprehensive research effort spanning a diverse range of tasks for the development and demonstration of polarization-based technology addressing some very challenging problems in Automatic Target Recognition and Detection (ATR/D). The scope of this effort involves extensive data collection, extension of polarization camera sensor design into the IR, physical reflectance and outdoor illumination modeling, rendering of predicted polarization signatures from physical models, algorithm development, and performance verification. The following summarizes some of the major aspects:

- Empirical collection of polarization imagery on various known domestic and foreign targets using visible, near-IR (night vision), and mid-IR (Forward Looking IR imaging), polarization sensing technologies.

- Accurate physical modeling of incident sky illumination, and, polarization reflectance from targets, camouflage, and terrain.

- Predictive rendering of characteristic target signatures using detailed BRL-CAD models of military target vehicles.

- Algorithm development, demonstration and evaluated performance using collected image data, guided by the science of reflectance modeling and predictive rendered signatures.

Empirical imaging of ground-based targets by ground-based polarization camera sensors will take place on test ranges at the Aberdeen Proving Ground, and, at Fort AP Hill. Therefore the environment in which targets will be embedded is primarily woodland terrain. Targets to be imaged will include both domestic and foreign assets, including United States M-1 Tank, M-3 Bradley Fighting Vehicle, HMMWV (High Mobility Multi-Wheeled Vehicle, of National Guard variety), and, Russian BMP-1 Infantry Fighting Vehicle, BTR70 Personnel Carrier, and, T72 Tank.

Three (3) basic types of polarization camera sensors will be used in three respective spectral ranges. These spectral ranges are the visible range 400-700nm, the near infrared range 700-900nm in which night vision photomultiplier technology operates, and, the mid-infrared ranges 3-5 micron and 8-12 micron. An important objective is to extend current polarization camera design that has worked so well in the visible spectrum, to the near-IR and mid-IR spectrums.

2.2 Some Details

Wavelengths in the visible, and near to mid IR spectrums are significantly smaller than any discernible shape detail on military targets and therefore polarization information provided at these wavelengths has very high spatial resolution. It is this characteristic that makes such polarization-based ATR/D algorithms advantageous for detecting and recognizing partially occluded targets, even possibly under very high occlusion. All that is required for detection/recognition is the line-of-sight visibility of a portion of a target with a discernible material and/or shape characteristic identified by a measured polarization signature, or, set of signtaures. This also applies to netted camouflage material and targets concealed by such camouflage. The following is a list of some of the related detailed objectives for this effort:

- Optimize detection and recognition performance of vehicle targets under various levels of occlusion by objects in woodland terrain, up to 90% occlusion (day and night operation).

- Optimize detection of camouflage nets in woodland terrain (day and night operation)

- For detected camouflage nets, optimize accuracy for detecting presence of a concealed vehicle (day and night operation)

- Optimize discrimination ability between actual targets and false decoys (day and night operation).

- Validate target and background polarization signatures with predictive reflectance modeling.

3 Research Issues

In the same vein as previous research efforts in polarization vision, the basis for algorithm development in polarization-based Automatic Target Recognition and Detection will have a strong grounding in the basic science of physics. The inherent philosophy for algorithm design is that polarization signatures used in algorithm development must be fundamentally predictable using physical models, otherwise robustness and repeatability for detection and/or recognition is tenuous.

3.1 Understanding Reflected and Thermally Emitted Polarization

Recent work in the visible spectrum [6] has shown a definitive correspondence between predicted reflected polarization signatures of vehicle targets from BRL-CAD models, and, empirical polarization imagery. Incorporated into renderings of polarization signatures are reflectance models as well as outdoor illumination polarization models. Clear skylight is not unpolarized– sunlight incident on atmosphere is polarized by scattering, the partial polarization dependent upon scattering angle according to Rayleigh's Law [1]. Further understanding is required for polarization signatures under a variety of outdoor weather conditions and this will require more accurate polarization illumination modeling respective to cloud cover and humidity/temperature conditions. One objective for algorithm development is to extend geometric flexible template matching for pose estimation (e.g., [2]) to include polarization radiometric features for both target recognition and pose estimation using such predictive modeling. A significant technical challenge is the ability to do this with as approximate knowledge of illumination conditions as possible. It is anticipated that predictive modeling of polarization signatures at night (i.e., using night vision photo-multiplication) will be considerably simpler, as starlight and moonlight produce very little atmospheric scattering and hence such light is unpolarized. Understanding polarization signatures from object vehicles at night in the 700-900nm range will be a significant research issue.

A major research issue in this effort involves understanding the polarization signatures respective to IR radiation particularly in the mid infrared 3-5 micron and 8-12 micron (day and night operation). Polarization in the mid-infrared (i.e., FLIR) for man-made objects offers even more signatures for enhanced ATR/D. While similar polarization effects occur for reflected infrared radiation, additional polarization phenomena occur for **thermal** radiation from heated objects. A model originally proposed for polarization of the diffuse component from objects in the visible spectrum using Fresnel theory [7] can be used to fundamentally predict thermally emitted polarization signatures. Part of this research effort is to determine what modifications and additions to this model are required for more precise prediction. Figure 1 shows a simulation of predicted partial polarization from a thermally emitting cylinder at 400°K predicted by the model in [7] revealing that significant partial polarization occurs where the local surface normal is very oblique relative to viewing (i.e., more than 60°). The implications of this are important. While highlighting and providing easier segmentation of occluding and self-occluding contours of heated portions of a vehicle (e.g., engine parts), recognition and detection of vehicles for night and day operation is significantly enhanced. This also provides a means for distinguishing a 3D thermal signature from a 2D thermal signature such as from electric blankets used to deceptively represent thermal emissions from engines on decoys for targets.

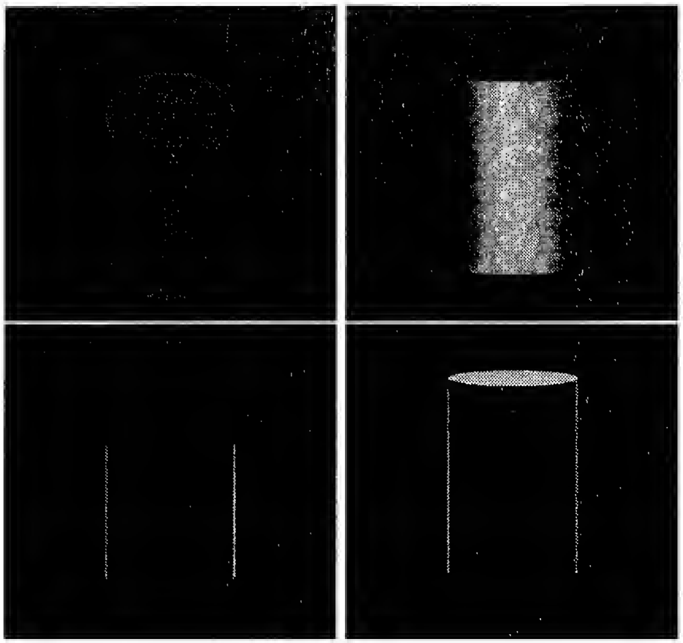


Figure 1: Two views of a simulated thermally emitting cylinder at 400° K. The top images are a standard shaded rendering of the cylinder to illustrate its pose. The bottom images render partial polarization from thermal emission as a grey value. Zero partial polarization is black. The brightest grey value in the bottom images corresponds to partial polarization of just above 40%. The cylinder on the left is tilted 45° forward from upright and the cylinder on the right is tilted 7° forward from upright. Note also how partial polarization becomes visible on the top surface as it becomes oblique.

Figure 2 shows an empirical polarization image (right) of a side view of a truck with engine running, using a 3-5 micron FLIR with polarizers installed in a filter wheel. This image was taken at the Lockheed-Martin facility, Denver Colorado. The left image shows a standard IR image while the right image of Figure 2 shows bright pixels superimposed on the darkened left image denoting where partial polarization from thermal emission is above 5%. This empirically demonstrates the same qualitative effect of high partial polarization present at surface normals significantly oblique relative to viewing, anticipated by the simulations shown in figure 1. The next step is quantitative comparison/verification from empirical IR polarization images of known ground-truth thermally radiating objects.

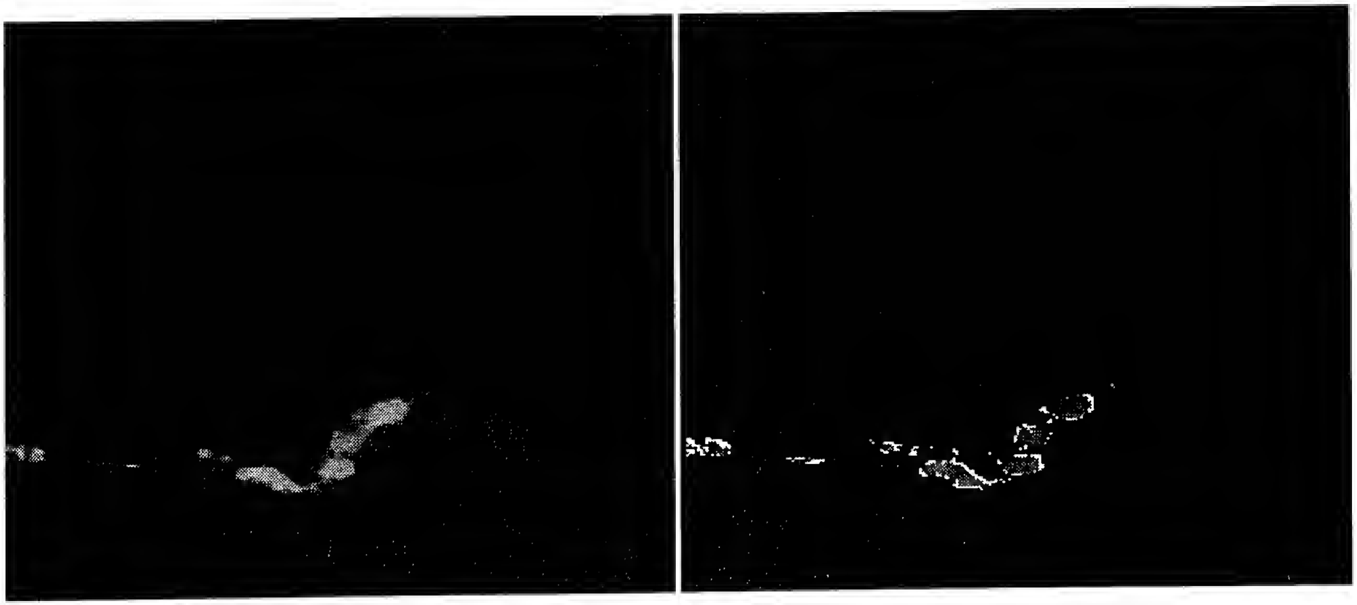


Figure 2: Two 3-5 micron FLIR images taken of the side of a truck with engine running, at Lockheed-Martin, Denver. The left image is a standard FLIR image. The right image is a polarization FLIR image which shows partial polarization greater than 5% in bright white superimposed on left image (left image is darkened for clarity of partial polarization depiction).

3.2 Extending polarization sensors into the IR

Under past DARPA efforts the Computer Vision Laboratory at Johns Hopkins has developed a polarization camera design for the broad visible spectrum using two twisted nematic (TN) liquid crystals and a fixed polarizer modularly fitted onto the lens of a CCD camera. The idea behind a liquid crystal polarization camera is very simple which is why it works so well. Nothing mechanically rotates; the polarizer remains fixed while the twisted nematic (TN) liquid crystals electro-optically rotate the plane of the linear polarized component of reflected partially linear polarized light in synchronization with the video rate of the camera. This is described in detail in [3, 4, 5]. A set of research issues to be solved early-on in this effort is to extend this design concept to sensors in the near-IR and mid-IR. One issue is the redesign of liquid crystals to accommodate the IR spectrum which is the primary issue for adaptation to night vision technology. A second issue which is primary to adaptation to FLIR is redesigning electronics for the liquid crystals so they do not interfere with the existing system. For both near IR and FLIR systems measurement calibration is another important issue.

4 Evaluation of Performance and Progress

The methodology for ATR/D algorithm performance evaluation will be to compute Receiver Operator Characteristic (ROC) curves for polarization image data subsets taken under specifically controled conditions. The ROC curves of interest to us will plot probability of target detection/recognition versus the number of false alarms for a given algorithm. A good performing algorithm will have high probability of detection with low false alarm rates.

Because this effort is effectively self-contained in obtaining its own data collections under controlled conditions, datasets can be obtained holding some variables constant (e.g., pixels on target at a given range and at a given percentage of occlusion) while varying other parameters (e.g., different viewing aspects). Currently planned are approximately 20-40 such polarization images to be taken for a given target under prescribed conditions and an ROC curve for particular algorithms can be computed. Now consider taking the dataset over again for a different level of occlusion and compute ROC curves for the same set of algorithms. How does the relative performance of each algorithm compare for different levels of occlusion ? Does this change from vehicle to vehicle ? The range of variables across which polarization image datasets will be collected includes: target type, range-to-target, pixels on target (or pixels on camouflage net), level of occlusion, orientation of target, spectrum the polarization image is in (i.e., visible, near-infrared, mid-infrared), sky illumination condition relative to viewing, weather condition, and whether it is day or night. Such a procedure identifies conditions under which certain algorithms perform well, and establishes their breaking point conditions.

Progress of this effort will be measured primarily in terms of improved empirical performance of ATR/D algorithms under different prescribed conditions. For instance, algorithms resulting from research that increases probability of recognition/detection while not increasing false alarm rates of recognition/detection for a given set of target conditions is one demonstration of a significant level of progress. Another level of progress can be measured by increased level of interest by the military in this technology for ATR/D applications as a result of demonstrated performance.

5 Conclusion

Overviewed in this paper is an effort for advancing the state-of-the-art in polarization based image understanding for Automatic Target Recognition and Detection. It was stressed that the effort will be highly empirical, guided by fundamental physical understanding of the processes that effect the polarization of light. Progress of this effort will be measured by demonstrated performance.

References

[1] S. Chandrasekhar. *Radiative Transfer.* Dover Publications, New York, 1960.

[2] R. Beveridge B. Draper and K. Siejko. Progress on target and terrain recognition research at colorado state university. *Proceedings of the DARPA Image Understanding Workshop*, pages 531–540, February 1996.

[3] L.B. Wolff. Polarization camera technology. In *Proceedings of the DARPA Image Understanding Workshop*, pages 1031–1036, Washington, D.C., April 1993.

[4] L.B. Wolff. Advances in polarization vision. *Proceedings of the DARPA Image Understanding Workshop*, November 1994.

[5] L.B. Wolff. Applications of polarization camera technology. *IEEE EXPERT*, 10(5):30–38, October 1995.

[6] L.B. Wolff. Reflectance modeling for outdoor object recognition and detection. *Proceedings of the DARPA Image Understanding Workshop*, pages 799–803, February 1996.

[7] L.B. Wolff and T.E. Boult. Constraining object features using a polarization reflectance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(7):635–657, July 1991.

Image Understanding Research at UC Irvine: Automatic Recognition in Multispectral Imagery

Glenn Healey

Computer Vision Laboratory
Electrical and Computer Engineering
University of California
Irvine, CA 92697
healey@ece.uci.edu
http://www.cvl.uci.edu

Abstract

In this project, we will develop and demonstrate new algorithms for the illumination and temperature invariant recognition of targets in multispectral infrared images. The algorithms will be based on the use of invariants computed from image regions. These invariants are derived from physical models for image formation and are independent of viewpoint and the illumination, atmospheric, and thermal environments. We have shown that invariants can be computed that capture arbitrary combinations of spectral and spatial information allowing spectral/spatial tradeoffs to be optimized according to the characteristics of a particular recognition problem. Since the algorithms are derived from physical models, constraints on the physical environment can be incorporated to improve performance. Extensive experiments will be conducted to demonstrate the effectiveness of the approach.

1 Introduction

Automatic target recognition systems which utilize only spatial information are severely limited in their ability to detect targets with low contrast or in environments containing camouflage or deception. The development of multispectral and hyperspectral infrared imagers promises to provide sensor input that can be exploited to improve the performance of target recognition systems in these circumstances. The use of multispectral infrared imagery for target recognition has several important advantages. Infrared sensors provide superior performance to other passive electro-optical imaging devices when night or bad weather operation is required. Infrared images provide information about both the reflective and emissive properties of objects in a scene. Exploiting this information across several bands can significantly improve the reliability of recognition especially in the presence of partial occlusion or camouflage.

For many years, multispectral image measurements have been used as a feature for recognition. Spectral measurements in satellite images, for example, are often used to classify areas of the earth according to known spectral signatures for classes such as water or vegetation. The matching of color distributions in visible images has been demonstrated for the recognition of many objects [14]. These methods work well in certain contexts, but depend on the assumption that spectral measurements for a given object do not change. Unfortunately, changes in illumination, atmospheric conditions, and the thermal environment can cause significant variation in observed spectral measurements for a fixed object. Consequently, the performance of methods which use direct spectral comparison for recognition degrades significantly in changing environments.

This project addresses the problem of automatic target recognition in multispectral infrared images. New recognition algorithms will be derived from a physical model for infrared image formation. The algorithms make use of invariants which capture spectral and spatial properties of objects but which are invariant to changes in illumination, atmospheric conditions, and temperature. A recognition system will be implemented based on invariant computation and indexing. The system will be evaluated by extensive ex-

This research is sponsored by DARPA and the Air Force Office of Scientific Research.

perimentation on multispectral images in several different environments.

2 Objectives

The algorithms developed in this project will significantly improve battlefield awareness in several areas. The algorithms exploit physical invariants allowing targets to be identified under an unprecedented range of conditions. Since the algorithms can capture arbitrary spectral and spatial characteristics of targets and backgrounds, they will improve on existing capabilities to discriminate camouflage from background and targets from decoys. The algorithms allow for targets to be characterized independent of context and for images to be illumination corrected for automatic verification of an identified target. After characterization, the algorithms can be used to track the target under a wide range of conditions. By using multispectral data to identify the material composition of a target, the algorithms can be used to match munitions to target composition. The algorithms can also be used for quantifying destruction in support of battle damage assessment.

The algorithms developed in this work can also be used to support the efforts of image analysts. The models can be used to characterize, identify, and track objects in terms of spectral and textural descriptors. Environmental parameters and physical descriptions of objects can be specified by the image analyst to constrain searches for objects of interest in multispectral images. Camouflage and decoys can be characterized and detected. Illumination effects can be compensated for to generate images which are easier to interpret. The algorithms can be used to identify materials in a scene for tasks such as determining terrain trafficability. Methods developed in this project can be used to register new imagery with previous imagery obtained under different illumination and atmospheric conditions.

3 Research Issues

3.1 Background

Color constancy is the ability to recover descriptors of an object from a color image that do not depend on the illumination. Human vision exhibits approximate color constancy over many different surfaces and illumination conditions. Color constancy is desirable for any recognition system that is required to function in environments where illumination cannot be controlled. For example, figure 1 displays images of the same textured surface under yellow (left) and red (right) illuminants. The sensed images are significantly different even though the underlying surface is the same. Computational approaches to color constancy have focused on the visible wavelengths where illumination variation is the most significant factor in causing variation in multispectral measurements for a fixed object.

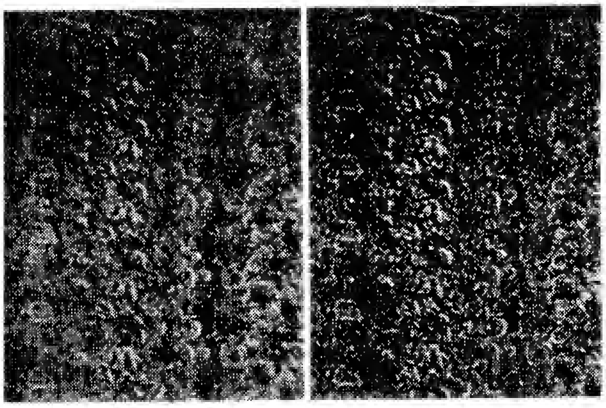


Figure 1: Surface under 2 illuminants

Without assumptions about the scene, color constancy is not possible [4]. Several algorithms for color constancy are based on the premise that spectral reflectance can be approximated by a linear combination of a fixed set of basis functions [3]. This premise has been carefully verified using large sets of visible spectral reflectance data [1] [7]. Recent work which uses linear reflectance models has combined spectral and spatial properties for illumination-invariant recognition using multispectral distributions [5] [11], local spatial structure [12], multispectral texture [6], and combined properties [2]. These techniques have been demonstrated for recognition on many visible multispectral images in the presence of large illumination changes. Some work has begun on finding invariants in single band long-wave infrared images [8]. At this stage, however, the physical basis for these invariants is still under investigation.

In this project, we will generalize computational color constancy methods to the infrared by developing algorithms for computing invariants from multispectral infrared images. Achieving this goal presents several important challenges. Infrared image measurements combine information about both reflected radiation and thermal emissions complicating infrared image modeling relative to the visible case where reflec-

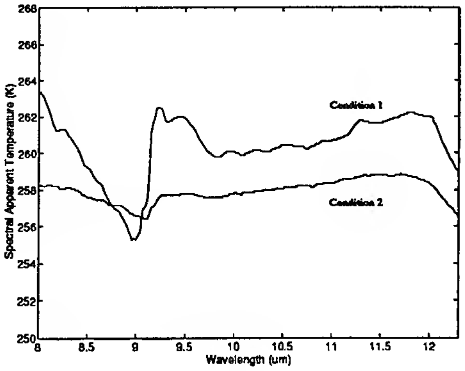


Figure 2: CARC-tan under two different conditions

tion is the dominant process. Infrared image measurements for specific objects will change with the ambient illumination, atmospheric conditions, and temperature. For example, figure 2 plots the measured spectral apparent temperature for a surface coated with tan chemical agent resistant coating (CARC-tan) under two different conditions [13]. The structure of the two functions is significantly different. The use of finite dimensional models for spectral reflectance has not been analyzed as carefully in the infrared as for visible wavelengths. Properties of these models must be understood to enable the computation of infrared invariants which are analogous to physical invariants which have been utilized in visible multispectral images. Multispectral infrared image sensors often provide many more spectral bands than visible sensors introducing computational issues of band selection and grouping. It is also desirable to develop a general approach for combining spectral and spatial information in a recognition system.

Our strategy for recognition is based on the computation of invariants from multispectral image regions. Using a physical model for infrared image formation and a linear model for spectral reflectance, we have shown that illumination and temperature changes correspond to affine coordinate transformations of multispectral distributions. From this relationship, we have derived a set of affine invariants that can be used for target recognition under a wide range of conditions. These invariants, however, do not exploit the spatial information in a multispectral image. It is possible, especially in environments containing camouflage and decoys, for regions with similar multispectral distributions to have subtle differences in spatial structure. In order to capture spatial information, we have shown that distributions in spatially filtered multispectral images deform according to a similar coordinate transformation in response to scene changes. Thus, affine invariants of filtered distributions are invariant to illumination, atmospheric conditions, and temperature. These filtered distributions capture information about spatial structure in an image according to the spatial properties of the filter used. Since these invariants can be tuned to capture desired spectral and spatial attributes, they can be optimized for many different target recognition problems.

3.2 Modeling IR Reflectance Functions

The infrared invariants described in 3.1 are based on the use of a linear model for spectral reflectance. An important aspect of this project is to determine regions of the infrared over which the finite dimensional spectral reflectance model is accurate. In this context, accurate means that the number of parameters required by the model matches the anticipated number of spectral bands. We have collected a large amount of high quality infrared spectral reflectance data to establish regions of the infrared over which the linear spectral reflectance model is accurate. Data for natural materials from $0.4 - 14\mu m$ has been obtained from the Remote Sensing Laboratory at Johns Hopkins University. This data includes measurements for various minerals, vegetation, rocks, and soils. Details of the measurement techniques are given in [9] and [10]. A database of spectral reflectance data for man-made materials has been obtained from the National Imagery Resource Library. These materials include concrete, road asphalts and tar, construction materials, paints, and roofing materials. The evaluation of this spectral reflectance data will follow a principal components analysis over subsets of the infrared as was performed by Maloney [7] for visible reflectance spectra. The study will provide guidance in the selection and grouping of infrared spectral bands for invariant computation. This analysis is also likely to be useful for other researchers working on algorithms for processing multispectral infrared image data.

3.3 Spectral/Spatial Tradeoffs

The multispectral distribution and spatial invariants provide a general set of tools for illumination and temperature invariant recognition in multispectral images. The invariants are derived from a general physical model for image formation and can be computed for any combination of spectral bands and spatial properties. Given this level of generality, an important aspect of algorithm development and evaluation will be to determine the most effective invariants for recognition. This will require the development of tools for optimizing the set of invariants chosen depending on generic or specific characteristics of targets, backgrounds, and the environment.

Existing multispectral and hyperspectral imagers provide from tens to hundreds of spectral measurements at each pixel. To optimize system efficiency, some subset of these bands should be selected for invariant computation. In addition, as long as the linear spectral reflectance model is satisfied, invariants can be computed for arbitrary subsets of bands. Thus, it is useful from a computational viewpoint to partition the selected bands into relatively small subsets for invariant computation. In particular, spectral bands which exhibit low correlation should be placed in distinct subsets. In this project, we will develop methods for band selection and grouping for target recognition problems.

An important issue in the design of a recognition system is the selection of the set of spatial filters which are used to generate the filtered images which are used to compute invariants. General considerations suggest the use of a set of filters which provides a compact representation that is descriptive enough to enable accurate recognition. Given the specifications for a recognition problem, optimized filter sets can be designed that maximize target/background discriminability while using a total number of invariants that is as small as possible. Another design consideration is the window size used for invariant computation. Larger windows provide better statistical distribution estimates, but are more susceptible to occlusion in the scene.

The spectral and spatial invariants must be simultaneously optimized and combined for recognition. Tradeoffs amongst spectral bands, spatial filters, and window size are interrelated. Increasing the number of spectral bands, for example, can often be used to reduce the window size required to achieve the same level of performance. We will develop methods for optimizing the spectral/spatial tradeoffs given characteristics of the recognition task. The goal will be to determine the most effective invariant features for recognition. Since different invariants differ in discriminatory power and estimated accuracy, we will derive optimal methods of combining and comparing vectors of invariants. The methods for invariant selection and combination will be evaluated using recognition results obtained on multispectral image data.

4 Evaluation

The proposed approach to target recognition is based on the use of invariants derived from physical models for multispectral infrared image formation. Since these invariants are computed directly from image regions, a recognition system will be implemented based on invariant computation and indexing. The models underlying the algorithms will be evaluated on large sets of spectral reflectance and spectral radiance data. The system itself will be evaluated over a wide range of targets and backgrounds under different conditions by experimentation with a large volume of multispectral image data. System performance will be compared to traditional multispectral recognition systems that are based on comparison of spectral signatures. This experimentation will allow us to quantify system performance for different classes of scenes and also to analyze systematically a number of spectral/spatial tradeoffs that impact performance.

4.1 Model Evaluation

A large collection of hyperspectral infrared radiance measurements has been assembled as part of the Joint Multispectral Sensor Program [13]. This data includes field measurements of various targets and natural backgrounds under different conditions during different seasons. This data will be used in conjunction with the spectral reflectance data described in 3.2 to determine the accuracy of the scene change model over regions of the infrared for various targets and backgrounds.

4.2 System Evaluation

We have obtained a large volume of multispectral image data for algorithm evaluation. The imagery has been collected as part of the Hyperspectral Digital Imagery Collection Experiment (HYDICE) and the Airborne Remote Earth Sensing Program (ARES). This data includes multispectral and hyperspectral images of targets and camouflaged targets in various environments including coastal, desert, forest, urban, and military.

Since the focus of this project is invariant recognition, the metric used for evaluation will be a comparison of the performance of the new algorithms against traditional multispectral approaches that use direct comparison of spectral signatures. The spatial and spectral information input to each system will be the same to enable a meaningful comparison. The results will be target recognition rates and false alarm rates for both systems as a function of spectral, spatial, and scene parameters. This experimentation will lead to better understanding of important issues as well as refinements in the models and algorithms. The experiments will allow us to characterize the set of scenes to which our algorithms may be applied reliably and to quantify the uncertainty in results for different classes of scenes. We expect to achieve a significant performance improvement over traditional approaches.

5 Summary

This research will advance understanding of the fundamental capabilities and limitations of multispectral infrared recognition systems which operate in changing environments. New algorithms will be developed for target recognition in the presence of unknown illumination, atmospheric conditions, and temperature. These algorithms are derived from a detailed physical model for infrared image formation and make use of invariants which can be computed for any combination of spectral bands and spatial properties. This flexibility enables these algorithms to be applied to a wide range of target recognition problems. Physical knowledge about targets, backgrounds, and the environment can be used in conjunction with the algorithms to improve performance and efficiency. This research will reveal the effectiveness and limitations of camouflage under different conditions. Insights gained during this work can be used to influence future data collection and sensor design. Efforts will be made to transfer the algorithms developed in this project to military and commercial systems.

References

[1] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychonomic Sci.*, 1:369, 1964.

[2] G. Healey and A. Jain. Retrieving multispectral satellite images using physics-based invariant representations. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8):842–848, August 1996.

[3] G. Healey and Q.-T. Luong. Color in computer vision: recent progress. In C.H. Chen, L.F. Pau, and P.S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*. World scientific, 1997.

[4] G. Healey, S. Shafer, and L Wolff, editors. *Physics-Based Vision: Principles and Practice, COLOR*. Jones and Bartlett, Boston, 1992.

[5] G. Healey and D. Slater. Global color constancy: recognition of objects by use of illumination-invariant properties of color distributions. *J. Opt. Soc. Am. A*, 11(11):3003–3010, November 1994.

[6] G. Healey and L. Wang. Illumination-invariant recognition of texture in color images. *J. Opt. Soc. Am. A*, 12(9):1877–1883, September 1995.

[7] L. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10):1673–1683, October 1986.

[8] N. Nandhakumar, J. Michel, D. Arnold, G. Tsihrintzis, and V. Velten. Robust thermophysics-based interpretation of radiometrically uncalibrated IR images for ATR and site change detection. *IEEE Trans. Image Proc.*, 6(1):65–78, January 1997.

[9] J. Salisbury and D. D'Aria. Emissivity of terrestrial materials in the 3-5 μm atmospheric window. *Remote Sensing of Environment*, 47:345–361, 1994.

[10] J. Salisbury, A. Wald, and D. D'Aria. Thermal-infrared remote sensing and Kirchhoff's law, 1. laboratory measurements. *Journal of Geophysical Research*, 99:11,897–11,911, 1994.

[11] D. Slater and G. Healey. The illumination-invariant recognition of 3D objects using local color invariants. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(2):206–210, February 1996.

[12] D. Slater and G. Healey. Using a spectral reflectance model for the illumination-invariant recognition of local image structure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–775, 1996.

[13] A. Stocker et al. Analysis of infrared hyperspectral measurements by the joint multispectral program. In *SPIE Volume 2469, Targets and Backgrounds: Characterization and Representation*, pages 587–602, Orlando, 1995.

[14] M. Swain and D. Ballard. Color indexing. *Int. J. Comp. Vision*, 7:11–32, 1991.

AUTOMATIC TARGET RECOGNITION (ATR) TECHNICAL PAPERS

Soft Competitive Principal Component Analysis Using The Mixture of Experts

Craig L. Fancourt and Jose C. Principe

Computational NeuroEngineering Lab
Dept. of Electrical Engineering, University of Florida, Gainesville, FL 32611
fancourt@cnel.ufl.edu

Abstract

A new algorithm is proposed that performs competitive principal component analysis (PCA) of an image. A set of expert PCA networks compete, through the Mixture of Experts (MOE) formalism, on the basis of their ability to reconstruct the original image. The result is that the network finds an optimal projection of the image onto a reduced dimensional space as a function of the input and, hence, of image content. As a by-product, the image is segmented and identified according to stationary regions.

1. Introduction

The optimal linear block transform for coding images is the Karhunen-Loeve transform (KLT), also known as principal component analysis (PCA). However, PCA assumes that the statistics of the image are stationary over the various sub-blocks, an assumption that is often violated in practice. One way around this is to use multiple PCA experts, and allow each expert to specialize on a different stationary region of the image.

Dony and Haykin [1995] developed just such a technique that uses multiple PCA experts that compete in an iterative, winner-take-all format. For each sub-block, the winning expert is the one whose sub-space projection has the greatest variance. In this sense, the individual PCA experts act as a set of matched eigenfilters. The winning expert is then granted the right to update its weights, using one of the on-line learning rules for which the weights are known to converge to the true eigenfilters.

However, hard competition is not without its difficulties. It is extremely sensitive to initialization, such that some experts may never be active. Dony and Haykin initialized all the experts to the global eigenfilters, plus a small random value, and found that this kept all experts active. They also placed the eigenfilters on a Kohonen [1982] neighborhood map, and annealed them down to hard competition.

However, hard competition may not be optimal. There may be sub-blocks of an image that can be reconstructed better using a combination of experts, in soft competition. In this paper, we propose to use the Mixture of Experts (MOE) as the soft competitive mechanism between expert PCA networks. Jordan and Jacobs [1994] first enticingly mentioned the possibility of using PCA experts in the MOE architecture but, to the best of our knowledge, no such application has appeared to date.

2. Principal Component Analysis

When $\boldsymbol{x}$ is a P dimensional zero mean random vector, it can be represented without error by the sum of P linearly independent vectors as

$$\boldsymbol{x} = \sum_{i=1}^{P} y_i \boldsymbol{w}_i = \boldsymbol{W}\boldsymbol{y} \tag{1}$$

The matrix $\boldsymbol{W}$ is deterministic and full rank. The columns of $\boldsymbol{W}$ span the P dimensional space and are called the basis vectors.

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_P \end{bmatrix} , \quad \boldsymbol{w}_i^T \boldsymbol{w}_j = \delta_{ij} \tag{2}$$

Since the vectors $\boldsymbol{w}_i$ satisfy an orthonormality constraint, the coefficients of the expansion can be found from:

Acknowledgment: this work was partially supported by NSF grant ECS-9510715.

$$y_i = w_i^T x \quad \Leftrightarrow \quad y = W^T x \tag{3}$$

Thus, y can be regarded as the output of a system that linearly rotates the input. Maximizing the variance of y_i over the data set, with respect to the weights, results in an eigenvalue equation for the autocorrelation matrix of the input:

$$Rw_i = \lambda_i w_i\,, \quad R = E[xx^T] \tag{4}$$

Starting from scratch, if we perform the expansion over a subset of the basis functions,

$$\tilde{x} = \sum_{i=1}^{M} y_i w_i = Wy \qquad M < P \tag{5}$$

then W is no longer full rank, and we can only approximate the input. However, it can be shown that minimizing the L2 norm of the reconstruction error over the data set,

$$C = E[\|x - \tilde{x}\|^2] \tag{6}$$

under the same orthonormality constraints as in (2), again results in the eigenvalue equation (4) but where the M eigenvectors are chosen on the basis of the M largest corresponding eiegenvalues. Thus, minimizing the L2 norm of the reconstruction error is the same as maximizing the output variance. The minimum cost in (6) is given by the sum of the discarded eigenvalues:

$$C_{min} = \sum_{i=M+1}^{P} \lambda_i \tag{7}$$

An alternative solution for PCA can be found by means of a linear system using adaptive algorithms. There are several such algorithms, however we will concentrate on Sanger's [1989] rule, for which the weight update after the presentation of the nth pattern is

$$\Delta W^T(n) = \eta\{y(n)x^T(n) - LT[y(n)y^T(n)]W^T(n)\} \tag{8}$$

where η is a learning rate parameter and LT denotes lower triangular. Equation (8) is an on-line algorithm to compute the optimal sub-space projection.

3. The Mixture of Experts

It is well known that, given a data set {X,D}, where X is the input, D is the desired signal, and a transfer function Y=f(X,W), where W are a set of free parameters, the maximum likelihood solution for W under a Gaussian assumption on the error is equivalent to minimizing the mean square error. However, many data sets do not exhibit single Gaussian modes. One example is data sets that include multiple valued targets.

The Mixture of Experts directly attacks this problem with a set of experts moderated by a gate, and modeling their errors as a mixture of Gaussian probability density functions (pdf's):

$$p(d|x) = \sum_{k=1}^{K} P(k|x)p(d|x,k) = \sum_{k=1}^{K} G_k(x)p(d|x,k)\,, \tag{9}$$

$$p(d|x,k) = \frac{\exp\left[-\frac{1}{2}(y_k - d)^T \Sigma_k^{-1}(y_k - d)\right]}{(2\pi)^{a/2}|\Sigma_k|^{1/2}} \tag{10}$$

where y_k is the output of the kth expert, d is the desired signal, a is the dimension of y. In order to reduce the number of free parameters, it is common to make the assumption that the cross correlation of the error of the kth expert can be approximated by

$$\Sigma_k = E[(y_k - d)(y_k - d)^T] \approx I\sigma_k^2 \tag{11}$$

The mixing coefficients, G_k, are a function of the input only, and can be regarded as prior probabilities. Multiplying (9) by the desired response, and integrating, we see that:

$$\begin{aligned} y &= E[d|x] = \int d \cdot p(d|x)d(d) = \\ &\sum_{k=1}^{K} G_k(x) \int d \cdot p(d|x,j)d(d) = \sum_{k=1}^{K} G_k(x)y_k(x) \end{aligned} \tag{12}$$

Thus the output of the network is a weighted sum of the expert's outputs, with the weighting coefficients provided by the gate.

The experts and gate can be chosen to be either linear or non-linear networks, although in the original formulation they were restricted to generalized linear models (GLIM's). Weigend [1995] used nonlinear experts and gate, and called the model nonlinear gated experts. The advantage of using nonlinear networks is that a single hierarchical layer can theoretically solve any problem. In this paper we will use linear experts and a non-linear gate.

Instead of maximizing the likelihood, it is customary to minimize the log-likelihood, resulting in the cost function over the entire data set:

$$C = -\ln L = \sum_{n=1}^{N} \sum_{k=1}^{K} G_k(n) p(d(n)|x(n), k) \qquad (13)$$

3.1 Gradient Descent Learning

The free parameters of the system are the weights of the gating network, the weights of the expert networks, and the cross correlation.

In order to insure that the outputs of the gating network sum to one, the output is a layer of softmax axons,

$$G_k = \frac{\exp(s_k)}{\sum_{j=1}^{K} \exp(s_j)} \qquad (14)$$

in which case the error backpropagated to the input side of the softmax layer is:

$$\frac{\partial C}{\partial s_k} = \sum_{n=1}^{N} G_k(n) - H_k(n) \qquad (15)$$

$$H_k \equiv \frac{G_k p(d|x, k)}{\sum_{j=1}^{K} G_j p(d|x, k)} \, . \qquad (16)$$

The H_k are posterior probabilities, in the sense that they depend on both the input and the outputs of the experts.

There is an analytical solution for the variances which, given the assumption in (11), is given by

$$\frac{\partial C}{\partial \sigma_k} = 0 \rightarrow \sigma_k^2 = \frac{\sum_{n=1}^{N} H_k(n) \|e_k(n)\|^2}{\sum_{n=1}^{N} H_k(n)} \qquad (17)$$

The weight change of some parameter, w_k, in the k^th^ expert network is proportional to the gradient:

$$\frac{\partial C}{\partial w_k} = \sum_{n=1}^{N} \frac{H_k(n)}{\sigma_k^2} \left[e_k(n) \frac{\partial}{\partial w_k} y_k(n) \right] \qquad (18)$$

In gradient descent, it is common to approximate the total gradient over all patterns by the instantaneous gradient. We now seek to expand (18) for the case when the experts are PCA networks.

4. The Mixture of Experts and Principal Component Analysis

We now propose for the first time to use linear PCA networks as the experts in the MOE formalism. In this paper, we restrict ourselves to gradient descent learning. We can incorporate Sanger's rule into the MOE by noting that the term in brackets in (18) is the weight update for an isolated network trained under gradient descent. Therefore, the weight update for the kth PCA expert is

$$\Delta W_k^T(n) = \eta \frac{H_k(n)}{\sigma_k^2} \{ y(n) x^T(n) - LT[y(n) y^T(n) W^T(n)] \} \qquad (19)$$

After training, segmentation can be achieved by applying a winner-take-all threshold to the gating network's output:

$$Class(n) = ArgMax_k[G_k(n)] \qquad 1 \le k \le K \qquad (20)$$

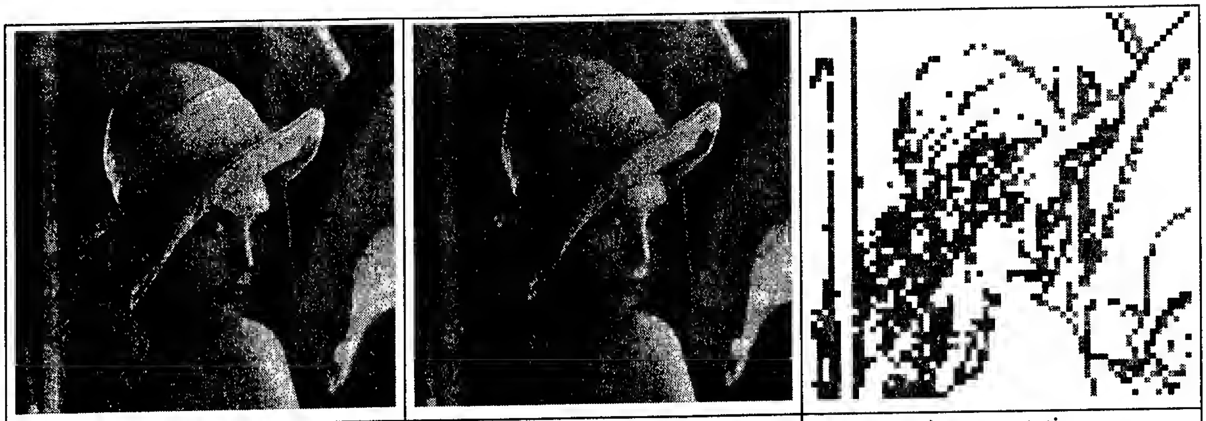
4.1 Training

We applied the model to a 512x512 grayscale image of Lena. The training set consisted of 4096 non-overlapping sub-blocks of 8x8 pixels. The mean was removed from each sub-block prior to training.

There were four linear PCA experts, each consisting of a 64 to 4 dimensionality reduction (and a 4 to 64 expansion for reconstruction). The gating network was a two hidden layer multi-layer perceptron, with four hyperbolic tangent activation functions on each hidden layer. The overall gate architecture was thus 64-4-4-4.

During training, the sub-blocks were randomly presented for approximately 100 epochs, until both the likelihood reached a plateau, and the segmentation stabilized.

The results are shown in Figure 1. The original



a) original image

b) reconstructed image

c) segmentation

Figure 1. Lena training results.

image is shown in a), the reconstructed image in b), and the segmentation in c). The segmentation is grayscale coded to indicate the winning expert, using (20), for each sub-block. White is used for the most frequent winning expert, while black is used for the least frequent winning expert.

Figure 2 below shows the masks of the four experts. The experts are ordered from most active (expert 1) to least active (expert 4), and the eigenvectors are ordered from the largest (top) to smallest (bottom) corresponding eigenvalues.

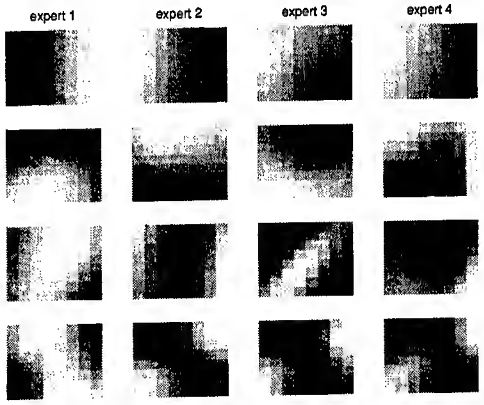


Figure 2. PCA masks of the four experts.

From Figures 1c and 2, we see that each expert clearly specialized on a particular feature of the image, since the experts' components are significantly different from each other. Expert 1 specialized on texture, while experts 2 and 3 specialized on edges.

It is clear that the MOE network is not only doing data reduction but also segmentation. This is an important observation since conventional PCA treats the entire image as a unique class, i.e. it is limited to the representation of the image. With our scheme each PCA component can represent a subset of the image patterns. Competition is utilized to find each piece, and thus the method brings discrimination into PÇA analysis, which has not been done in the past.

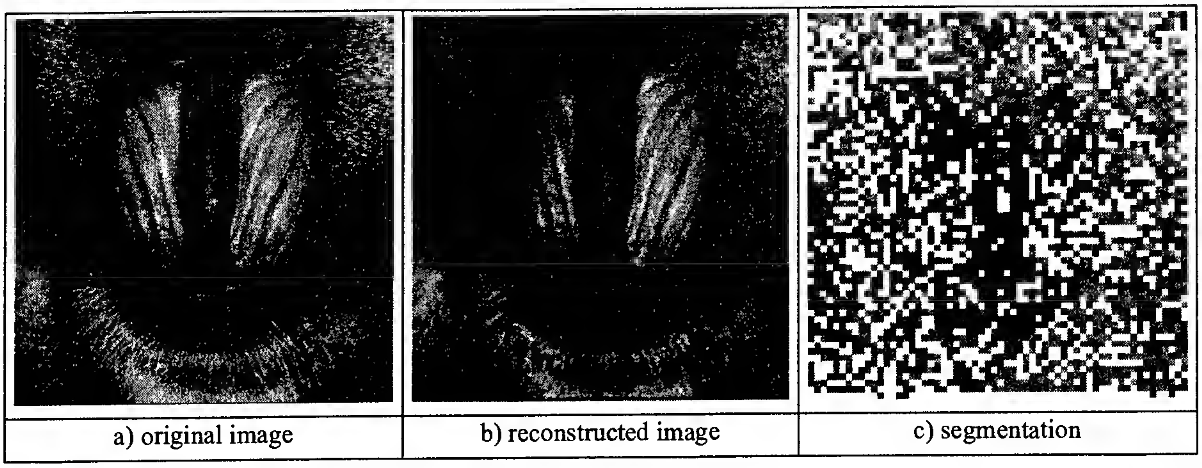
4.2 Generalization

We tested the trained model on a 480x480 grayscale image of a mandrill. The results are shown in Figure 4 on the following page. We see that the reconstruction is reasonable, but that the segmentation is not nearly as smooth. This is most likely due to the predominance of high frequency regions, which demonstrates that, to be universal, the method needs to be trained on more images.

4.3 Comparison with the global KLT

We also compared the mean squared error (mse) performance of the mixture of PCA experts to a global KLT. We did two tests. In the first, we made the KLT reduction 64:4, identical to each of the experts. The KLT reconstructed image for this case is shown in Figure 3 below. In the second, we kept the number of free weights the same, resulting in a KLT reduction of 64:16. If the method was strictly hard competition, the first case would be appropriate. However since the gate is a softmax function, all of the experts can contribute to the reconstruction of each sub-block, although in practice, most sub-blocks were dominated by one or two experts.

The results for both the Lena and mandrill



a) original image | b) reconstructed image | c) segmentation

Figure 3. Mandrill testing results.

images are shown in Table 1.

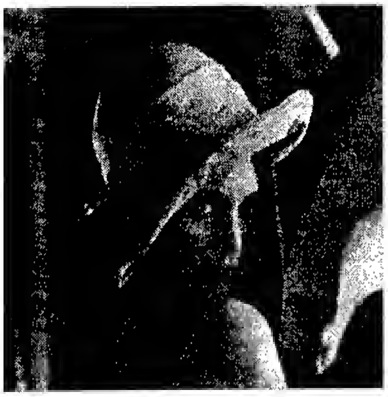


Figure 4. Global KLT reconstruction (64:4).

From the table we see that, for the Lena image, the MOE did outperform the global KLT with the same number of basis. However, it did worse when compared to the overall number of free PCA weights. This is what we would expect, but we believe that the MOE performance can be improved (these are preliminary results). For the case of the mandrill, the MOE performed reasonably well considering that the KLT performance was determined using the mandrill's own global eigenvectors.

Table 1: MSE ($\times 10^{-3}$) Comparison.

Model\Image	Lena	Mandrill
MOE (64:4)x4	1.05	11.5
KLT 64:4	1.08	10.8
KLT 64:16	0.26	5.1

5. Conclusions

This paper extends for the first time the MOE formalism to PCA analysis. The importance of this extension is that competition is brought into PCA analysis, which has been lacking. This may remove the major obstacle of applying PCA for signal classification. We also show that the MOE performs a little better than PCA with the same number of basis, but further improvements are possible.

References

Dony R., and Haykin S., Optimally Adaptive Transform Coding, *IEEE Transactions on Image Processing*, vol. 4, no. 10, 1995.

Haykin S., Neural Networks: A Comprehensive Foundation, New York: Macmillan, 1994.

Jacobs R.A., Jordan M.I., Nowlan S.J., and Hinton G.E., Adaptive mixtures of local experts, *Neural Computation*, vol. 3, pp. 79-87, 1991.

Jordan M.I., and Jacobs R.A., Hierarchical Mixtures of Experts and the EM Algorithm, *Neural Computation*, vol. 6, pp. 181-214, 1994.

Kohonen T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

Sanger T., Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Networks, vol. 2, pp. 459-473, 1989.

Weigend A.S., Mangeas M., and Srivastava A.N., Nonlinear gated experts for time series: discovering regimes and avoiding overfitting, International Journal of Neural Systems, Vol. 6, No. 4, 1995.

A Nonparametric Methodology for Information Theoretic Feature Extraction

John W. Fisher III
fisher@cnel.ufl.edu

José C. Principe
principe@cnel.ufl.edu

Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL 32611
http://www.cnel.ufl.edu

Abstract

We present a new information theoretic approach for training a system in a self-organized or supervised manner. Information theoretic signal processing looks beyond the second order statistical characterization inherent in the linear systems approach. The information theoretic framework probes the probability space of the data under analysis. This technique has wide application and represents a powerful new advance to the area of information theoretic signal processing.[1]

1.0 Introduction

We have recently developed a new information theoretic technique for feature extraction [Fisher and Principe, 1995]. This method differs from previous methods in that it is not limited to linear topologies [Linsker, 1988] nor uni-modal probability density functions (PDFs) [Bell and Sejnowski, 1995]. The method is directly applicable to any nonlinear mapping which is differentiable in its parameters. In particular, we demonstrate that the technique can be applied to a feed-forward multi-layer perceptron (MLP) with an arbitrary number of hidden layers.

Our goal is a theory and methodology to extract optimal features from observations of the data.

2.0 Information Theoretic Perspective

We approach the feature extraction problem from an information theoretic perspective. Within this framework, the feature extraction mapping is viewed as a special form of a communications channel, $g(\alpha,\ )$, which includes nonlinear subspace projections. The design goal is to maximize the *information* transmitted through the channel.

In this approach there are two basic assumptions. First, we assume that we have a finite set of observations of the data under consideration. Second, since the method addresses pattern recognition we assume that the data lies in a subspace.

The figure of merit we use is mutual information which can be written

$$I(x, y) = h(y) - h(y|x), \tag{1}$$

where $h(\ )$ is the differential entropy of a continuous random variable [Papoulis, 1991],

$$h(x) = -\int f_X(x)(\log(f_X(x)))dx, \tag{2}$$

and $h(y|x)$ is the conditional entropy which substitutes conditional probabilities.

The appeal of mutual information as a criterion for feature extraction is threefold. First, mutual information exploits the structure of the underlying probability density function. Adaptation, as we will show, can be used to remove as much uncertainty about the input, x, using observations of the output, y. Third, this is accomplished within the constraints of the mapping topology.

One obstacle to using mutual information as the figure of merit is that it is an integral function of

[1] this work was partially supported by DARPA grant F-33615-97-1-1019

the PDF of a continuous random variable. Since we cannot work with the PDF directly (unless assumptions are made about its form), we rely on nonparametric estimates. Nonparametric density estimation in a high-dimensional space is an ill-posed problem. The approach described here, however, relies on such estimates in the output space, as depicted in figure 1, where the dimensionality is under the control of the designer.

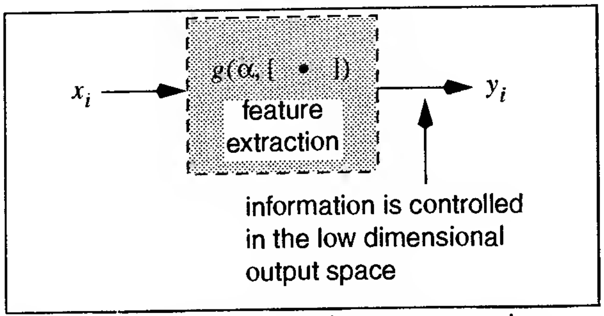


Figure 1. Mapping as feature extraction. Information content is measured in the low dimensional space of the observed output.

Viola *et al* [1996] has taken a very similar approach to entropy manipulation, although that work differs in that it does not address arbitrary nonlinear mappings directly, the gradient is estimated stochastically, and entropy is manipulated explicitly.

3.0 Derivation of the Learning Algorithm

As we stated, our goal is to find features that convey maximum information about the input. How do we adapt the parameters, α, of a mapping such that this is the case?

Consider the mapping $g:\Re^N \to \Re^N$; $M < N$, of a random vector $X \in \Re^N$, which is described by the following equation

$$Y = g(\alpha, X) \tag{3}$$

If $g(\alpha, x)$ is linear, very little can be done to manipulate the information about X contained in Y without prior knowledge of the pdf of X [Deco and Obradovic, 1996]. A nonlinear mapping, however, can exploit the following property of entropy. If a random variable has a finite region of support, its entropy is maximized when the random variable is uniformly distributed throughout the region of support. One candidate topology which a restricted output range, among other desirable properties, is the MLP using sigmoidal nonlinearities. Furthermore, if the range of the mapping is a hypercube (as in the MLP) the elements of the output vector are statistically independent.

By changing the parameters of the mapping such that the observed *output* distribution is uniform, we have effectively maximized entropy (i.e. information). In fact, by defining an appropriate distance metric, we can use the method described here to maximize or minimize entropy depending on the desired goal. As a consequence of our choices the approach fits naturally into a back-propagation learning framework.

As our criterion we use integrated squared error between our estimate of the output distribution, $\hat{f}_Y(u, y)$ at a point u over a set of observations y, and the desired output distribution, $f_Y(u)$, which we approximate with a summation.

$$\begin{aligned} J &= \frac{1}{2}\int_{\Omega_Y} (f_Y(u) - \hat{f}_Y(u, y))^2 du \\ &\approx \sum_j \frac{1}{2}(f_Y(u_j) - \hat{f}_Y(u_j, y))^2 \Delta u \end{aligned} \tag{4}$$

In equation 4, Ω_Y indicates the nonzero region (a hypercube for the uniform distribution) over which the M-fold integration is evaluated. Assuming the output distribution is sampled adequately, we can approximate this integral with a summation in which $u_j \in \Re^M$ are samples in M-space and Δu is represents a volume.

We use the Parzen window method [Parzen, 1962] as our estimator of the output distribution. The Parzen window estimate of a PDF is written

$$\hat{f}_Y(u, y) = \left(\frac{1}{N_y}\right) \sum_{i=1}^{N_y} \kappa(y_i - u), \tag{5}$$

where $\kappa(\ )$ is the kernel function, $y = \{y_1, \ldots, y_{N_Y}\}$ are the set of observations at the output of the mapping, and u is the location at which the output estimate is being computed. Since the output observations are functional mappings of the input data, we can rewrite 5 as

$$\begin{aligned} \hat{f}_Y(u, y) &= \left(\frac{1}{N_y}\right) \sum_{i=1}^{N_y} \kappa(g(\alpha, x_i) - u) \\ &= \hat{f}_Y(u, g(\alpha, x)) \\ &= \left(\frac{1}{N_y}\right) \sum_{i=1}^{N_y} \kappa(g(\alpha, x_i) - u) \end{aligned} \tag{6}$$

The gradient of the criterion function with respect to the mapping parameters is determined via the chain rule as

$$\begin{aligned}\frac{\partial J}{\partial \alpha} &= \left(\frac{\partial J}{\partial \hat{f}}\right)\left(\frac{\partial \hat{f}}{\partial g}\right)\left(\frac{\partial g}{\partial \alpha}\right)\\ &= -\Delta u\left(\sum_j (f_Y(u_j) - \hat{f}_Y(u_j, y))\right)\left(\frac{\partial \hat{f}}{\partial g}\right)\left(\frac{\partial g}{\partial \alpha}\right), \quad (7)\\ &= \Delta u \sum_j \varepsilon_Y(u_j, y)\left(\frac{\partial \hat{f}}{\partial g}\right)\left(\frac{\partial g}{\partial \alpha}\right)\end{aligned}$$

where $\varepsilon_Y(u_j, y)$ is the observed distribution error over all observations y. The last term in 7, $\partial g / \partial \alpha$, is recognized as the sensitivity of our mapping to the parameters α. Since our mapping is a feed-forward MLP (α represents the weights and bias terms of the neural network), this term can be computed efficiently using standard backpropagation. The remaining partial derivative, $\partial \hat{f} / \partial g$, is

$$\begin{aligned}\frac{\partial \hat{f}}{\partial g} &= \left(\frac{1}{N_Y}\right)\sum_{i=1}^{N_Y} \kappa'(y_i - u_j)\\ &= \left(\frac{1}{N_Y}\right)\sum_{i=1}^{N_Y} \kappa'(g(\alpha, x_i) - u_j)\end{aligned} \quad , \quad (8)$$

where κ'( ) is the derivative of the kernel function with respect to its argument.

Substituting 8 into 7 yields

$$\begin{aligned}\frac{\partial J}{\partial \alpha} &= \left(\frac{\Delta u}{N_Y}\right)\sum_j \varepsilon_Y(u_j, y)\sum_i \kappa'(y_i - u_j)\frac{\partial}{\partial \alpha}g(\alpha, x_i)\\ &= \frac{1}{N_Y}\sum_i \frac{\partial}{\partial \alpha}g(\alpha, x_i)\Delta u \sum_j \varepsilon_Y(u_j, y)\kappa'(y_i - u_j)\end{aligned} \quad (9)$$

The terms in 9, excluding the mapping sensitivities, become the new error direction term in our backpropagation algorithm. By reversing the order of summations in 9 we see that the error direction term associated with each observation is a convolution of the estimated error in the observed output distribution, $\varepsilon_Y(u, y)$, with the gradient of the kernel, κ'( ). It is through the gradient of the estimator kernel that the distribution error influences the direction of each data observation and thereby (through backpropagation) the parameters of the mapping. This point will be further illustrated in the next section for the case of gaussian kernels.

This adaptation scheme is depicted in figure 2. As can be seen, this approach fits readily into the backpropagation framework. The point set $x = \{x_1, \ldots, x_N\}$ is mapped to a point set $y = \{y_1, \ldots, y_N\}$. The criterion then estimates from the set an error between the observed output distribution and the baseline output distribution (uniform in this case). From this distribution error computed over the range of the output space, an error direction (whose sign depends on whether we wish to minimize of maximize entropy) is associated with each data point in the set y. This error direction is then backpropagated through the MLP in order to modify the parameters of the mapping.

4.0 Gaussian Kernels

Examination of the gaussian kernel and its differential in two dimension illustrates some of the practical issues of implementing this method of feature extraction as well as providing an intuitive understanding of what is happening during the adaptation process. The N-dimensional gaussian kernel evaluated at some u is (simplified for two dimensions)

$$\begin{aligned}\kappa(\mathbf{u}) &= \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}}\exp\left(-\frac{\boldsymbol{u}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{u}}{2}\right)\\ &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}}{2\sigma^2}\right) \quad . \quad (10)\\ \Sigma &= \sigma^2 I, N = 2\end{aligned}$$

The partial derivative of the kernel (also simplified for the two-dimensional case) with respect to the input y_i as observed at the output of the MLP is

$$\frac{\partial \kappa}{\partial \boldsymbol{u}} = \kappa(\boldsymbol{u})\Sigma^{-1}\boldsymbol{u} . \quad (11)$$

These functions are shown in figure 3.

Recall that the term

$$\Delta u \sum_j \varepsilon_Y(u_j, y)\kappa'(y_i - u_j)$$

in 9 replaces the standard supervised error direction term in the backpropagation algorithm. From the figure we see that when we are maximizing entropy, the distribution error through the kernel functions act as local attractors computed PDF error is positive and as a local repellor when the PDF error is negative. When we are minimizing entropy the behavior is opposite. In this way the adaptation procedure operates in the feature space locally from a globally derived measure of the output space (PDF estimate).

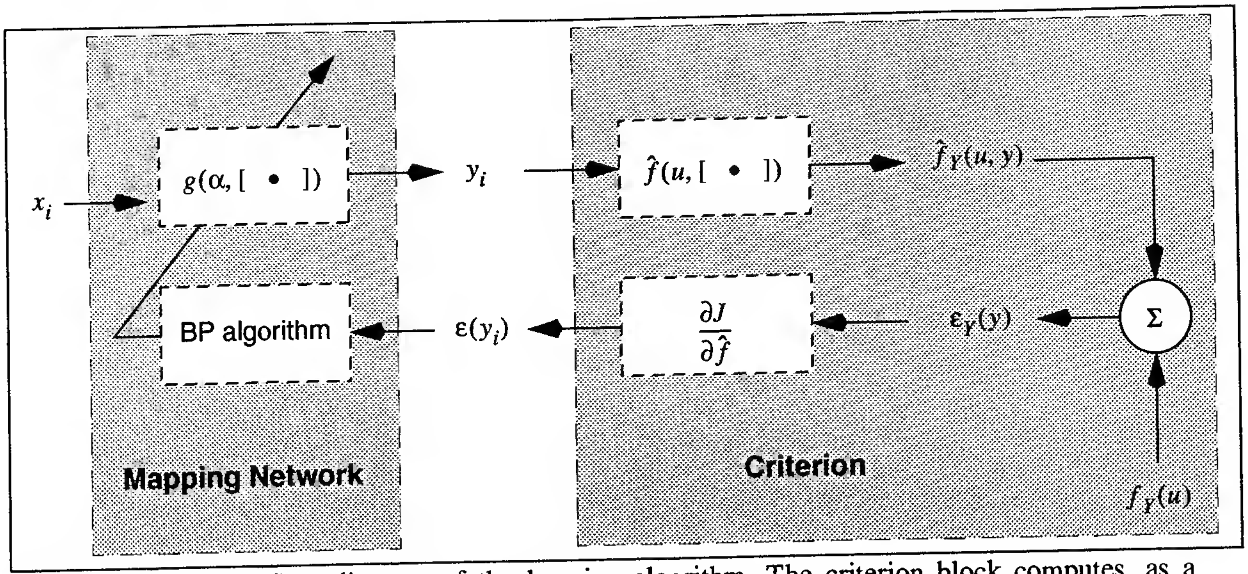


Figure 2. A signal flow diagram of the learning algorithm. The criterion block computes, as a function of the observed outputs, the error direction for the mapping network.

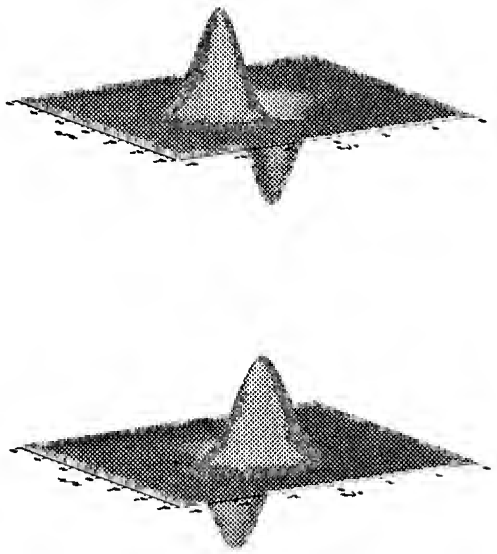


Figure 3. Gradient of two-dimensional gaussian kernel. The kernels act as attractors to low points in the observed PDF on the data when entropy maximization is desired.

5.0 Experimental Results

We present experimental results designed to illustrate the advantages of the information theoretic approach.

5.1 PCA/Entropy Comparison

In the first experiment we wish to illustrate the differences between the well known principal components analysis (PCA) approach to feature extraction as compared to an entropy driven approach. We will begin with the simple case of a two dimensional gaussian distribution. The distribution we will use is zero mean with a covariance matrix of

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

The contours of this distribution are shown in figure 4 along with the image of the first principal component features. We see from the figure that the first principal component lies along the x_0-axis. We draw a set of observations (50 in this case) from this distribution and compute a mapping using an MLP and the entropy maximizing criterion described in previous sections. The architecture of the MLP is 2-4-1, indicating 2 input nodes, 4 hidden nodes, and 1 output node. The nonlinearity used is the hyperbolic tangent function. We are therefore, nonlinearly mapping the two-dimensional input space onto a one-dimensional output space. The plot at the bottom of figure 4 shows the image of the maximum entropy mapping onto the input space. From the contours of this mapping we see that the maximum entropy mapping lies essentially in the same direction as the first principal components.

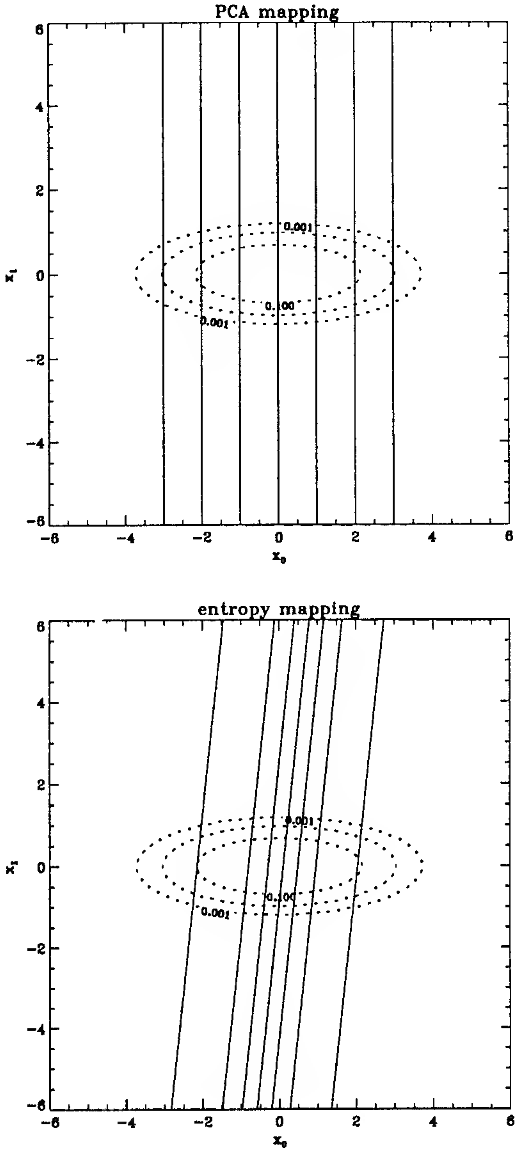


Figure 4. PCA vs. Entropy. Top: image of PCA features shown as contours. Bottom: Entropy mapping shown as contours.

This result is expected. It illustrates that when the gaussian assumption is supported by the data, maximum entropy and PCA are equivalent. This result has been recognized by many researchers.

We are also concerned with the case when the gaussian assumption is not correct. In which case we would not expect PCA and entropy direction to be equivalent. We conduct a second experiment to illustrate this point where we draw observations from a random source whose underlying distribution is not gaussian. Specifically the PDF is a mixture of gaussian modes with the following form

$$p(x) = 1/2(N(x, m_1, \Sigma_1) + N(x, m_2, \Sigma_2))$$

where $N(x, m, \Sigma)$ is a gaussian distribution with mean m and covariance Σ. In this specific case

$$m_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$$
$$m_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}.$$

It can be shown that the principal components of this distribution are the eigen vectors of the matrix

$$R = \frac{1}{2}(\Sigma_1 + m_1 m_1^T + \Sigma_2 + m_2 m_2^T)$$

This distribution is shown at the top of figure 5 along with its first principal component feature mapping. The bottom of figure 5 shows the image of the maximum entropy mapping. As we can see there are two distinct differences between this mapping and the PCA result. The first observation is that the mapping is nonlinear. The second observation is that the maximum entropy mapping is more tuned to the structure of the data in the input space.

This experiment helps to illustrate the differences between PCA and information (entropy). PCA is primarily concerned with direction finding and only looks at the second order statistics of the underlying data, while entropy explores the structure of the data class. In a few limited cases, second order statistics are sufficient (e.g. gaussian) to describe such structure, but in general they are not.

5.2 A Simple Classification Problem

The next experiment illustrates the difference between the linear systems and information theoretic approaches to classification. We begin with a two class problem in which the underlying distribution for class one is gaussian with mean m_1 and standard deviation σ_1 while class two is gaussian with mean zero and standard deviation $\sigma_1/2$. It is well known that the optimal discriminant function chooses class one if $x < -1.29$ or $0.62 < x$. The two distributions, are shown in the top plot of figure 6.

In the bottom plot of the same figure we replace class one with an exponential distribution of the same mean and standard deviation as class one.

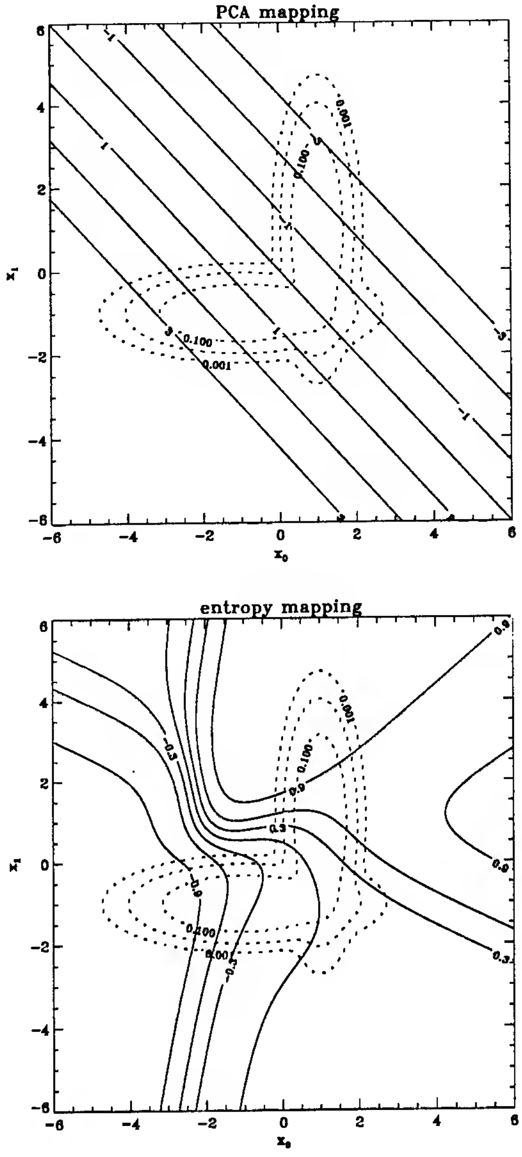


Figure 5. PCA vs. Entropy. Top: image of PCA features shown as contours. Bottom: Entropy mapping shown as contours.

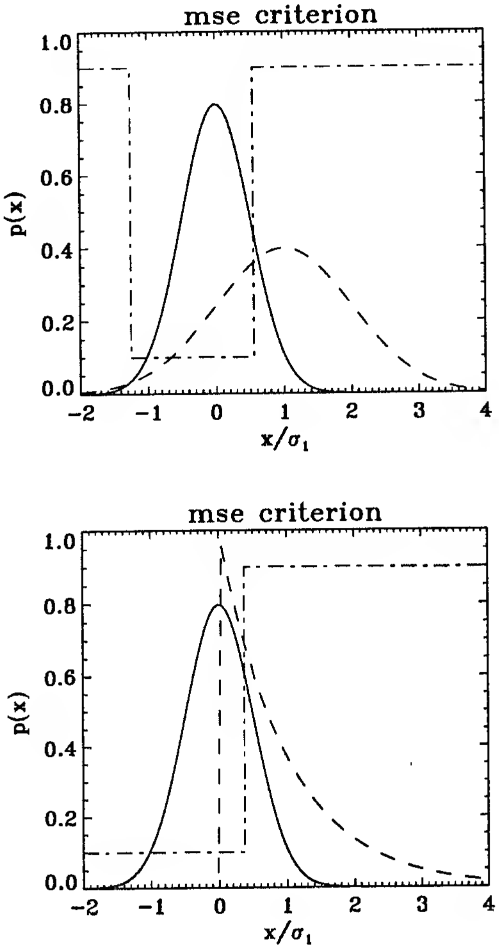


Figure 6. Two class problem. Case 1 (top): both distributions are gaussian. Case 2 (bottom): one distribution is gaussian, the other is exponential. The discriminant thresholds are shown as a dashed-dotted line.

The optimal discriminant function in this case chooses class one if $x > 0$.

Both of these discriminant functions can be represented with MLPs. The question is how to adapt the parameters of the mapping. The usual approach is to assign a functional value to either class and use the mean-square error (MSE) criterion to adapt the weights of the MLP. When this is done the resulting discriminant functions are shown for both cases overlaid onto the distributions in figure 6. From the figure we can see that in the first case (both classes are gaussian), the resulting discriminant function is close to the optimal, however, in the second case, the discriminant is significantly biased from the optimal solution. We attribute this result to that fact that the MSE criterion by itself is not sufficient to examine the underlying structure of the input data.

Alternatively we first compute a maximum entropy mapping over the mixture class. After remapping the input data we use the same MLP as before to compute a new discriminant function. In the gaussian case the result is not substantially different, however, as shown at the top of figure 7, the result for the gaussian/exponential case has a much better result. The discriminant function is now much closer to the optimal solution than when the MSE criterion alone was used. We attribute this result to the fact that the maximum entropy pre-processor

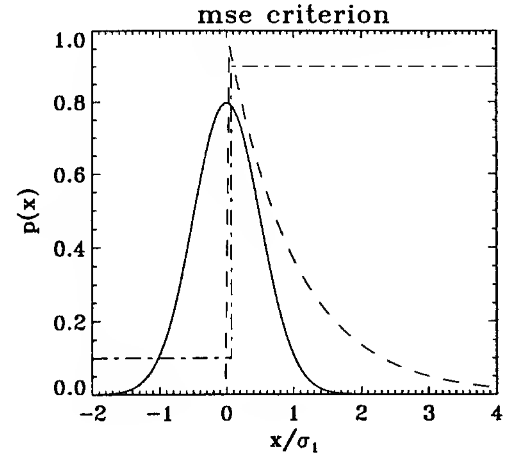


Figure 7. Entropy/MSE criterion result. Gaussian versus exponential. Entropy preprocessing of the mixture class removes the bias in the discriminant threshold (dashed-dotted line).

maps the input data such that the subsequent discriminant is easier to achieve in the MLP architecture.

6.0 Conclusions

We have described a nonparametric approach to information theoretic feature extraction. We believe that this method can be used to improve classification performance by directly choosing relevant features for classification. A critical capability of the information theoretic approach is the ability to adapt the entropy of the output space of the nonlinear projection entropy conditioned on the input data. We have shown that through the use of a simple differentiable estimator, namely Parzen windows, that the adaptation of entropy can fit logically into the error backpropagation model. This method is different from other entropy based approaches such as using the Kullback-Leibler norm for supervised learning.

We have also presented experiments that illustrate the usefulness of this technique. Comparisons to the well known PCA method show that the information theoretic approach is more sensitive to the underlying data structure beyond simple second-order statistics.

We have also shown that the approach improved the resulting discriminant function for a non-gaussian classification problem. The data types used for the experiments were simple by design. They served to illustrate the usefulness of the method even for seemingly simple problems. We believe that for more complicated data structures such as SAR imagery the improvement will be more significant. We will be reporting such results in the future.

References

Bell, A., and T. Sejnowski (1995); “An Information-Maximization Approach to Blind Separation and Blind Deconvolution”, *Neural Computation* 7, 1129-1159.

Deco, G., and D. Obradovic (1996); *An Information-Theoretic Approach to Neural Computing*, Springer-Verlag, New York, inc.

Fisher J., and Principe, J. C. (1995); “Unsupervised learning for nonlinear synthetic discriminant functions”, *Proceeding of SPIE*, vol. 2752, 1-13.

Linsker, R. (1988); “Self-organization in a perceptual system”, *Computer*, 21, 105-117.

Papoulis, A. (1991), *Probability, Random Variables, and Stochastic Processes* (3rd ed.), McGraw-Hill, Inc.

Parzen, E. (1962); “On the estimation of a probability density function and the mode”, *Ann. Math. Stat.* 33, 1065-1076.

Viola, P., N. Schraudolph, and T. Sejnowski,(1996); “Empirical Entropy Manipulation for Real-World Problems”, *Neural Information Processing Systems* 8, to appear in published proceedings.

Toward a Fundamental Understanding of Multiresolution SAR Signatures

Gilbert Leung and Jeffrey H. Shapiro

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
and Research Laboratory of Electronics
Cambridge, Massachusetts 02139 USA

ABSTRACT

A physical optics formalism is used to establish a first-principles analysis for discriminating specular returns from diffuse returns in a 1-D synthetic aperture radar. The optimum Neyman-Pearson detection processor is shown to substantially outperform the conventional, full-resolution SAR imager for extended specular targets.

1. INTRODUCTION

Synthetic aperture radars (SARs) provide the coverage rate and all-weather operability needed for wide-area surveillance. SAR-based automatic target recognition (ATR) systems need fast and effective discriminators to suppress vast amounts of natural clutter from, while admitting the much more limited set of man-made object data to, their classification processors. Recent research, using mm-wave SAR data, has demonstrated that multiresolution processing offers a useful discriminant in this regard.[1] Other work, with ultra-wide-band foliage-penetrating SAR data, has shown that adaptive-resolution imaging can exploit the aspect-dependent reflectivity of man-made objects.[2] Neither these studies, nor other related work,[4] have taken a principled approach—one based on the physical characteristics of the reflecting surfaces and SAR operation—to multiresolution SAR image formation and optimal target detection. The present paper is a first step toward such a fundamental assessment.

Using the physical optics formalism established in Park and Shapiro,[3] we consider multiresolution SAR image formation for a 1-D SAR, i.e., a continuous-wave downlooking sensor. We find that the carrier-to-noise ratios (CNRs) for diffuse and specular reflectors have different multiresolution signatures. Thus, although a diffuse reflector and a specular reflector of the same size have identical normalized CNRs when their SAR returns are processed over the full dwell time, these reflectors show substantially different behavior when processed over shorter time intervals. In particular, the "broad-side flash" phenomenon exploited by Chaney et al.[2] is clearly present in our specular CNR analysis. This specular CNR behavior directly impacts the structure and performance of the Neyman-Pearson optimal detector for such a reflector: for extended specular targets the optimum detection processor substantially outperforms the conventional, full-resolution SAR imager.

2. 1-D SAR PRINCIPLES

The essential principles that permit multiresoution imaging to distinguish specular returns from diffuse returns can be gleaned from simple physical arguments, as we now show.

2.1. Doppler Pulse-Compression Imaging

Consider the simple continuous-wave (cw) downlooking radar imager sketched in Fig. 1. A plane flying at speed v and height L above the ground collects returns, i.e., reflections, from the scatterers it irradiates. Because of the plane's motion, the Doppler-shift time history associated with the return from a point scatterer located at $\vec{r}_s$ is a linear function—a frequency chirp—of rate $\dot{\nu}_D = -2v^2/\lambda L$, with zero-intercept at x/v, where x is the along-track component of $\vec{r}_s$, and λ is the radar's operating wavelength. The time duration of the chirp from this single scatterer is the dwell time, i.e., the length of time during which the particular scatterer lies within the radar's transmitter beam, given by $T \approx \lambda L/vd$ in diffraction-limited far-field operation with an antenna of diameter d. By analogy with pulse compression operation of an angle-angle imager, we can show that matched-filter (chirp-compression) processing of the return from this single point scatterer will yield a time-domain output waveform that is centered at x/v and has a nominal width x_{res}/v, where

$$x_{res} \approx \frac{v}{|\dot{\nu}_D|T} \approx \frac{d}{2} \ll \frac{\lambda L}{d}, \tag{1}$$

gives the along-track spatial resolution of the system.

Send correspondence to J.H. Shapiro.
Email: jhs@mit.edu; Phone: 617-253-4607; Fax: 617-258-7354.

Research supported by U.S. Air Force Office of Scientific Research Grant F49620-93-1-0604.

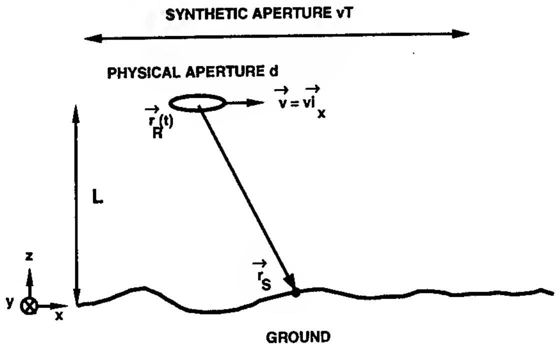


Figure 1. Geometry for a 1-D continuous-wave synthetic aperture radar.

2.2. Diffuse versus Specular Reflections

The simple, point-scatterer decription we have given for the resolution capability of a 1-D SAR applies directly to returns from an extended rough surface, i.e., to a diffuse reflector, but it does not apply directly to the returns from an extended smooth surface, i.e., a specular reflector. As sketched in Figs. 2 and 3, the returns from each point on a rough surface add incoherently, whereas the returns from each point on a smooth surface add coherently. In particular, the rough surface shown in Fig. 2 will produce a total return whose time duration equals the time this surface spends within the radar's transmitter beam. On the other hand, the smooth surface shown in Fig. 3 will produce a total return whose time duration is determined by the time it takes the radar's receiver antenna to move through the diffraction pattern created by the reflected wave. For a large specular target this time duration will be much less than the dwell time T given in the previous section. Furthermore, because of Snell's law, the tilt of such a specular target will lead to a time offset in the return field arriving at the radar receiver.

3. SYSTEM MODEL AND CNR

In order to quantify the phenomena described qualitatively in the preceding section, we shall extend the physical optics 1-D SAR analysis of Park and Shapiro[3] to include specular, as well as diffuse, reflectors, and multiresolution, as well as full-resolution imaging.

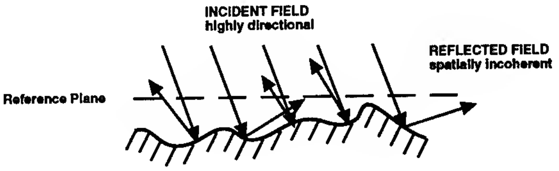


Figure 2. Rough surfaces give diffuse reflections.

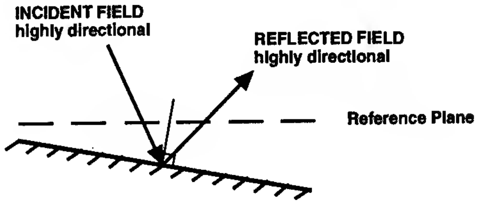


Figure 3. Smooth surfaces give specular reflections.

3.1. IF Signal Model

The intermediate-frequency (IF) signal in the SAR receiver can be taken to have complex envelope $\mathbf{r}(t) = \mathbf{y}(t) + \mathbf{w}(t)$, where $\mathbf{y}(t)$ represents the return waveform and $\mathbf{w}(t)$ is a zero-mean circulo-complex, white Gaussian receiver noise with spectral density N_0. In a convenient normalization, and assuming a multiplicative target model,[3,5] the return complex envelope can be written in the following form:

$$\mathbf{y}(t) = \sqrt{P_T}\int d\vec{\rho}\, T(\vec{\rho})\xi_L^2(\vec{\rho} - \vec{v}t), \qquad (2)$$

where P_T is the transmitter power, $T(\vec{\rho})$ is the field-reflection coefficient vs. the transverse coordinate vector $\vec{\rho} \equiv (x, y)$ on the $z = 0$ reference plane, and $\xi_L(\vec{\rho})$ is the normalized transmitter field pattern on this reference plane. In this equation, we have assumed that the radar's transmitter and receiver use the same antenna pattern, that lag angle compensation has been performed. We shall assume that ξ_L results from L-m far-field free-space propagation of an elliptical Gaussian beam whose major and minor axes coincide with the across-track and along-track directions, respectively. The choice of a Gaussian pattern is for analytical convenience; the use of an elliptical pattern allows the across-track resolution to gain the superior resolution of a larger aperture dimension while the along-track resolution benefits from synthetic aperture processing.[3] Our model for a simple extended diffuse reflector is statistical: $T(\vec{\rho})$ is a zero-mean, circulo-complex Gaussian random field with a δ-function covariance,

$$\langle T(\vec{\rho})T^*(\vec{\rho}')\rangle = \frac{\lambda^2\rho_d}{\pi}\exp(-2|\vec{\rho}|^2/a^2)\delta(\vec{\rho} - \vec{\rho}'), \qquad (3)$$

where ρ_d is the diffuse reflectivity and a is the radius of the reflecting region. Our corresponding model for a plane mirror of the same nominal size is essentially deterministic:

$$T(\vec{\rho}) = \sqrt{T_s}\exp(-|\vec{\rho}|^2/a^2 + j2k\vec{\theta}\cdot\vec{\rho} + j\phi_s), \qquad (4)$$

where T_s is the intensity reflectivity, $\vec{\theta}$ is the tilt, ϕ_s is the phase shift of the mirror, and $k \equiv 2\pi/\lambda$ is the wave number at the radar wavelength. Because we seldom know the radar-to-ground distance to a small fraction of a wavelength, we shall assume that ϕ_s is a random variable that is uniformly distributed on $[0, 2\pi)$; the other parameters in our specular model are non-random. We have used Gaussian shapes in our diffuse and specular models because they will allow closed-form solutions for multiresolution CNR and other calculations while preserving the essential physical parameter dependencies of more realistic shape functions.

3.2. Multiresolution Carrier-to-Noise Ratio

The architecture of a simple 1-D SAR receiver is shown in Fig. 4. It consists of a chirp-compression filter—whose integration interval that has zero time-offset, duration T_c—followed by a video detector. The carrier-to-noise ratio of this receiver, CNR$(t; T_c)$ is defined to be the mean target-return power in the cw-SAR image at time t divided by the mean receiver-noise power in the cw-SAR image at time t. We are interested in seeing whether or not the qualitative differences between the diffuse and specular reflectors, described earlier, are indeed present in the IF signal model we have posited. Specifically, does CNR$(t; T_c)$ behave differently, for the diffuse and specular reflectors, as the processing time T_c is increased from very small values up to the full dwell time T?

Using the model given in the previous section, we find that the diffuse target's CNR obeys

$$\mathrm{CNR}(t; T_c) = K_d \rho_d T_c \exp\left[-\frac{(vt)^2}{\Delta x_d^2}\right], \tag{5}$$

and the specular target's CNR is given by,

$$\mathrm{CNR}(t; T_c) = K_s \frac{k^2 a^2 T_s T_c}{2} \exp\left[-\frac{(vt - x_s)^2}{\Delta x_s^2}\right], \tag{6}$$

where K_d and K_s are constants that do not depend on the processing time T_c. These equations show that the Doppler-compressed diffuse and specular returns have CNRs with different spatial extents Δx_d and Δx_s, respectively, and that the Doppler-compressed specular return has a spatial offset x_s that the corresponding diffuse case does not. These forms are indicative of the signatures we had hoped for: different resolutions (t-widths of CNR$(t; T_c)$) for the specular and diffuse returns arising from the coherent nature of the specular reflection process, and an offset for the specular return that is due to Snell's law. Figures 5 and 6 show that the desired signatures do, in fact, exist. In drawing these figures we have assumed that: the radar's receiver is in the far field of the specular target; the radar's integration interval is long compared to the aperture-translation time and short compared to the target-dwell time; and the target is much larger than the along-track aperture size of the radar transmitter.

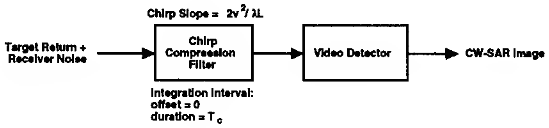


Figure 4. Block diagram of 1-D SAR receiver.

Figure 5 shows that the diffuse-target resolution follows

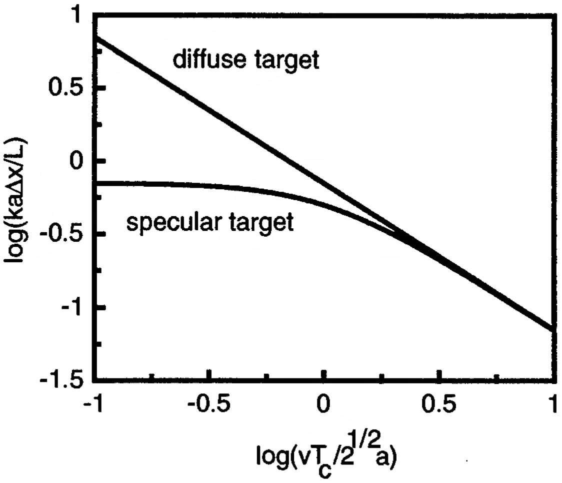


Figure 5. CNR comparison: specular vs. diffuse target resolution.

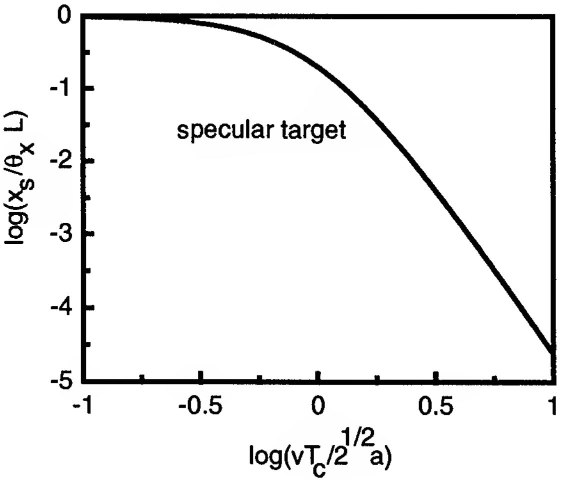


Figure 6. CNR comparison: specular target angular shift.

a straight reciprocal dependence on the processing time. This is to be expected. We have assumed, in Fig. 5, that T_c is large enough to form a significant synthetic aperture, but small enough that the full, dwell-time limited synthetic aperture is not used. The more interesting part of Fig. 5 is the specular target behavior. At short processing times the resolution is limited by the target's size, and independent of T_c. Ultimately, the synthetic aperture effect takes over, and the specular curve merges with the diffuse curve. For our equal-sized specular and diffuse targets, this means their full-resolution (dwell-time limited processing) images will have the same Δx values. Figure 6 shows the aspect dependence of the specular return. Here we see that geometric optics (Snell's law) behavior prevails at short processing times, but the tilt dependent shift in the SAR image disappears as the processing time is increased.

4. TARGET DETECTION THEORY

Our final task will be to show that the multiresolution signatures demonstrated in the last section have important implications for target detection. We consider the following idealized binary hypothesis testing problem. Under hypothesis H_0, the normalized IF complex envelope satisfies,

$$\mathbf{r}(t) = \mathbf{y}_d(t) + \mathbf{w}(t), \tag{7}$$

where $\mathbf{y}_d(t)$ is the return from a statistically homogeneous diffuse clutter, i.e., $\mathbf{y}_d(t)$ is given by Eq. 2 for a diffuse reflector whose covariance function is found from Eq. 3 with $a \to \infty$. Under hypothesis H_1, we have that

$$\mathbf{r}(t) = \mathbf{y}_s(t) + \mathbf{y}_d(t) + \mathbf{w}(t), \tag{8}$$

where $\mathbf{y}_s(t)$ satisfies Eq. 2 with $T(\vec{\rho})$ from Eq. 4. Under either hypothesis, $\mathbf{w}(t)$ is white Gaussian receiver noise.

The optimum Neyman-Pearson detection processor for this problem processes $\mathbf{r}(t)$ to obtain maximum detection probability $P_D \equiv \Pr(\text{ say } H_1 \mid H_1 \text{ true})$ for a prescribed value of the false-alarm probability $P_F \equiv \Pr(\text{ say } H_1 \mid H_0 \text{ true})$. Note that we will assume that this processor knows all the parameters of the diffuse and specular reflectors. (This same assumption will be made for the two suboptimum receivers described below.) In a real detection scenario, many of these parameters may be unknown and will have to be estimated from the data. Our desire here is to show that the multiresolution processor can offer significant performance improvement over a conventional, full-resolution SAR imager. If this is not the case in an idealized, known-parameter setting, it seems unlikely that any such advantage could prevail in more realistic situations.

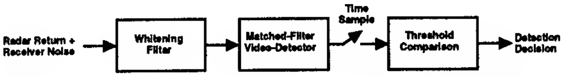


Figure 7. Optimum Neyman-Pearson receiver.

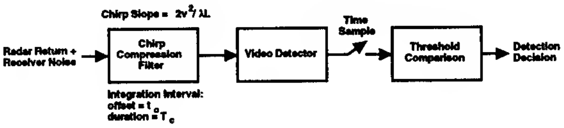


Figure 8. 1-D SAR imager receivers.

The optimum detection problem we have posed is well known.[6] The structure of the optimum receiver is shown in Fig. 7; it consists of a whitening filter followed by a matched-filter/video-detector and then a threshold test. There are two other receivers whose performance we shall compare to that of the optimum system: the conventional, full-resolution 1-D SAR imager, and the optimized multiresolution 1-D SAR imager. These receivers share a common block diagram, shown in Fig. 8. The difference between the these two receivers is as follows. The conventional imager uses no integration-interval offset in its chirp compressor. In other words, it makes no attempt to accommodate the Snell's law shift in the time at which the specular-target return occurs. Also, the conventional imager uses the full dwell time for its chirp compression integration; it does not try to exploit the multiresolution specular signature we demonstrated in the previous section. In contrast, the optimized multiresolution imager chooses its integration-interval offset and duration to maximize the resulting P_D at the prescribed P_F. This receiver represents an idealized, known-parameter 1-D SAR version of the aspect-dependent processor reported by Chaney et al.[2]

The receiver operating characteristics (P_D vs. P_F behaviors) of the preceding three receivers share a common form,[6]

$$P_D = Q(d, \sqrt{-2\ln P_F}), \tag{9}$$

where

$$Q(\alpha, \beta) \equiv \int_{\beta}^{\infty} dz\, z \exp(-\alpha^2/2 - z^2/2) I_0(\alpha z), \tag{10}$$

is Marcum's Q function, and I_0 is the zeroth-order modified Bessel function. In Eq. 9, d^2 is the effective signal-to-noise ratio (SNR), which, in general, has a different value for each of our three receivers. We have plotted Eq. 9 for several d values in Fig. 9. The rest of our analysis shall concentrate on the behavior of the effective SNRs.

Let us use d_o^2, d_c^2, and d_w^2 to denote the effective SNRs for the optimum receiver, the conventional full-resolution

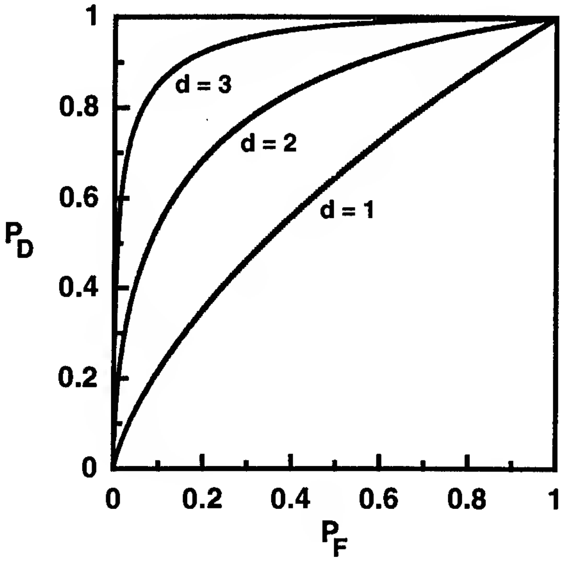


Figure 9. Marcum's Q function

1-D SAR receiver, and the optimum white-noise receiver, respectively. The optimum white-noise receiver achieves maximum P_D at the allowed P_F for our binary hypothesis test when there is no diffuse clutter present under either hypothesis. It turns out that this receiver takes the form of an optimized multiresolution SAR imager in the absence of clutter, and is very nearly the optimized multiresolution SAR in the presence of clutter.

We first consider the case in which tilt is not an issue, i.e., $\vec{\theta} = \vec{0}$ in our specular target model. Figures 10 and 11 show log-log plots of the optimum receiver's effective SNR advantages (d_o/d_w and d_o/d_c), as a function of normalized target size (a/d with d being the radar's along-track antenna diameter), for two values of the diffuse-return (clutter) carrier-to-noise ratio, CNR_d.

By definition, we have $d_w \to d_o$ as $\text{CNR}_d \to 0$. Figure 10 shows that for extended specular targets, i.e., for $2a/d > 1$ this equivalence prevails even when the clutter is strong. Physically, the large specular target presents a shorter-than-dwell-time return to the radar receiver. Hence, this return has a narrower bandwidth—less frequency chirp—than the clutter return. The performance of the white-noise receiver approximates that of the optimum receiver because the clutter spectrum is nearly flat over the bandwidth of the return from the extended specular, and hence the whitening filter in the optimum receiver is superfluous. Figure 11 shows that the optimum recevier has many decibels of effective SNR advantage over the conventional receiver for a large specular target ($a/d \gg 1$). The conventional receiver collects noise over the full chirp bandwidth of the dwell time, and this

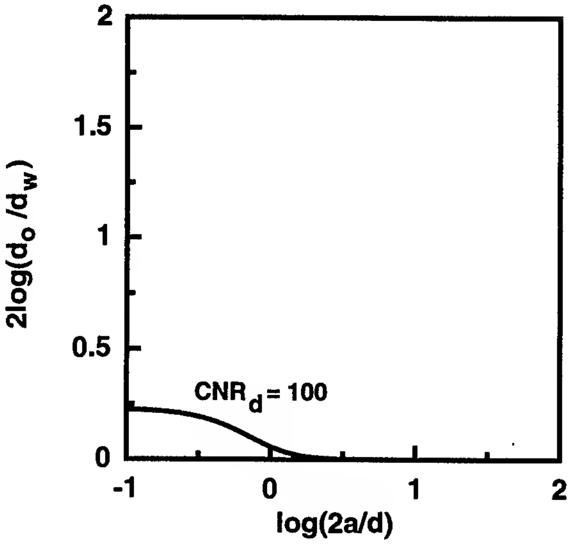


Figure 10. Effective SNR comparison: optimum receiver vs. white-noise receiver at zero target tilt. The $\text{CNR}_d = 0.1$ curve is indistinguishable from the $d_0 = d_w$ line.

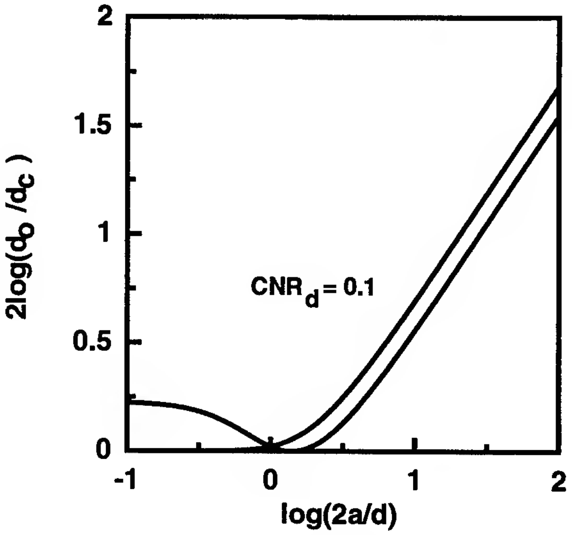


Figure 11. Effective SNR comparison: optimum receiver vs. conventional receiver at zero target tilt.

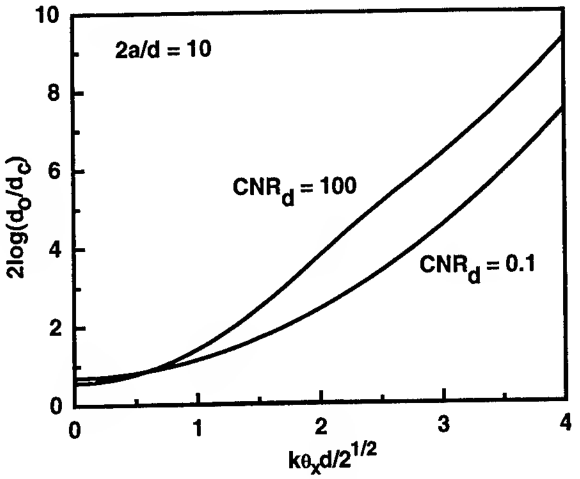


Figure 12. Effective SNR comparison: optimum recevier vs. conventional receiver as a function of along-track target tilt.

extra noise drives its SNR down, relative to that of the optimum receiver, because the optimum receiver uses the much narrower bandwidth associated with the more limited chirp present on the specular target return.

Our final example addresses the impact of target tilt. In Fig. 12 we have plotted d_o/d_c vs. normalized along-track tilt at $2a/d = 10$ for two values of CNR$_d$. Note that $d_o \approx d_w$ prevails at this value of $2a/d$, so this figure also constitutes a comparison between conventional and optimized multiresolution SAR imagers. These curves underline the value of exploiting the aspect-dependence of the return from a large specular target.

5. CONCLUSIONS

Our continuing objective is to develop a principled approach to the use of multiresolution image formation for discriminating specular returns from diffuse returns in synthetic aperture radar data. In this our initial effort, we have used a simple cw 1-D SAR model to establish the fundamental validity of using multiresolution, aspect-dependent specular target effects for the discrimination task. Complete derivations of our results are given in Leung,[5] where the more general specular-reflector case of an elliptically-symmetric curved mirror is considered. This source also includes target models and CNR behaviors for dihedral and trihedral reflectors, as well as a comparison of the structure and performance of optimum Neyman-Pearson, conventional 1-D SAR, and optimized multiresolution SAR receivers for the detection of a finite diffuse target embedded in white Gaussian receiver noise. Our current work includes the extension of our formalism to 2-D stripmap operation, to polarimeteric SAR, and to the detection and recognition of multicomponent targets.

REFERENCES

1. W.W. Irving, A.S. Willsky and L.M. Novak, "A Multiresolution Approach to Discriminating Targets from Clutter in SAR Imagery," Proc. SPIE **2487**, 272–299 (1995).
2. R.D. Chaney, A.S. Willsky, and L.M. Novak, "Coherent Aspect-Dependent SAR Image Formation," Proc. SPIE **2230**, 256–274 (1994).
3. D. Park and J.H. Shapiro, "Performance Analysis of Optical Synthetic Aperture Radars," Proc. SPIE **999**, 100–116 (1988).
4. N.S. Subotic, B.J. Thelen, J.D. Gorman, and M.F. Reiley, "Multiresolution Detection of Coherent Radar Targets," IEEE Trans. Image Process. **6**, 21–35 (1997).
5. G. Leung, "Synthetic Aperture Radar Discrimination of Diffuse and Specular Target Returns," M.Eng. thesis, Dept. of Elect. Eng. and Comput. Sci., MIT, Feb. 1997.
6. H.L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, Wiley, New York, 1968, Chap. 4.

Use of Context for False Alarm Reduction in SAR Automatic Target Recognition

S. Kuttikkad W. Phillips S. Mathieu-Marni R. Meth
R. Chellappa
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

This paper introduces techniques for context-aided false alarm reduction for Automatic Target Recognition (ATR) in airborne Synthetic Aperture Radar (SAR) images. Candidate target chips are identified using Constant False Alarm Rate (CFAR) detection techniques. If only a single image of a site is available, a 2-D site model is constructed and used to determine regions inhospitable to targets. A framework for the registration and exploitation of multipass imagery is developed. After registration, these images are used to provide consistency-based false alarm reduction. A final discrimination step separates surviving false alarms from targets. Experimental results using the TESAR and ADTS datasets are included.

1 Introduction

A high-resolution airborne radar operating at nonfoliage penetrating frequencies and moderate depression angles can produce images of targets, in the clear or under partial occlusion. Bright scatterers in radar imagery due to targets can be detected using CFAR processing [Rohling, 1983], where each pixel is compared to an adaptive threshold that is a function of the desired probability of false alarm, a statistic derived from the reference clutter window around the pixel, and the size of this reference window.

Cardinal streaks and other bright returns from buildings and associated rooftop substructures and false alarms from foliage increase the burden on ATR algorithms. This paper describes several techniques for false alarm reduction using context, prior to discrimination, for SAR ATR. The context information could be derived from a single image using segmentation, or from registered multipass imagery. Given a single radar image of a site and the result of a target detection algorithm, it is possible to focus the attention of ATR algorithms by delineating buildings, dense foliage, and other areas where targets are not expected to be found. Alternately, if multipass imagery of a site is available, false alarms can be eliminated by checking for consistency in detector output across images. Although different statistical models for the clutter lead to different CFAR detectors, the algorithms presented here are independent of the type of detector chosen.

In Section 2, we revisit the problem of deriving context from a single SAR image by constructing a 2-D site model. A multiresolution technique for segmenting SAR intensity images under Weibull clutter assumptions is presented. A maximum-likelihood segmentation is performed at the coarsest resolution and the region labels and a confidence measure are propagated to finer resolutions. Only pixels with low confidence are reclassified, yielding a smoother segmentation map with a smaller computational burden.

Two SAR images of the same area are projections of the three dimensional (3-D) scene onto different slant planes. Hence, a Euclidean or similarity transformation is not sufficient to register the two images. In Section 3, we show that registration of two SAR images can be approximated by an affine transformation which consists of the following sequential steps: projection of the first image to the ground plane; rotation and translation within the ground plane; and projection to the slant plane of the second image. This transformation can be derived from the sensor acquisition parameters, with a post-registration refinement for translational errors. One application for exploiting registered SAR imagery, namely computing heights of buildings and the foliage canopy,

The support of the Air Force Office of Scientific Research and the Defense Advanced Research Projects Agency (ARPA Order No. A369) under Grant F49620-93-1-0576 is gratefully acknowledged.

is developed and applied to real multi-pass airborne SAR imagery from MIT Lincoln Laboratory.

Section 4 describes schemes for false alarm reduction in single and multipass SAR imagery with examples. After context-aided false alarm reduction, a discrimination step, described in Section 5, is used to further narrow down the candidate targets. Discrimination is performed with the help of size and shape features derived from the target chips.

2 Multiresolution Segmentation and Region Labeling

In earlier work [Chellappa *et al.*, 1996; Kuttikkad and Chellappa, 1995; Kuttikkad *et al.*, 1996], we have described techniques for generating 2-D site models from single and multipolarization SAR imagery. Our site model consists of delineations of buildings, roads, and possible target clusters (including military and civilian vehicles), as well as natural regions such as trees, shadows, grass, etc. The algorithm consists of three stages—bright pixel detection, segmentation, and labeling/recognition. Bright scatterers in nonhomogeneous clutter are detected using CFAR processing. Next, a maximum likelihood (ML) segmentation into a small number of expected terrain classes is performed to label the image. The conditional distribution of the backscatter, given the region label, is chosen appropriately based on the type of data available (single/multiple looks, intensity/complex, single/multipolarization, etc.). At typical depression angles, a building produces a bright linear or L-shaped streak along its edge(s) facing the sensor, followed by a shadow region. Bright pixel clusters are identified in the CFAR output and checked for elongated streaks by fitting rectangles of a minimum aspect ratio and length, and testing for shape conformity. Detected streaks are then combined with shadow information from the ML labeling stage to identify buildings. Shape constraints are used to aggregate road regions into road segments. Finally, since the operating frequency of the sensor does not permit foliage penetration, tree/grass ambiguities are resolved using supporting shadow evidence.

To improve the critical ML segmentation stage of the site model construction algorithms, we have developed a multiresolution technique for segmenting single-polarization SAR intensity images. The motivation for the development of a multiresolution algorithm is smoother, more accurate segmentation to facilitate context extraction and region-adaptive detection. For instance, existing segmentation algorithms often misclassify pixels at tree-grass boundaries in high-resolution imagery. Misclassifications are caused because tree edges are very similar to grassy regions and are interspersed with shadows that lower the first-order statistics into a range closely matching the grass class. The effect is usually a ring of incorrectly labeled grass pixels surrounding regions of tree canopy.

Most of the previous multiresolution work (such as [Krishnamachari and Chellappa, 1997] and its references) is designed for texture segmentation and is not appropriate for the speckled images produced by SAR. Fosgate [Fosgate *et al.*, 1997] developed a SAR multiresolution segmentation algorithm for two-class (tree/grass) segmentation of complex data. Linear prediction across scales is used to classify the fine-resolution pixels. An ML technique based on the distributions of the prediction error residuals has been effectively used to solve the two-class problem. A problem with this technique is the refinement procedure needed to obtain accurate labeling at boundaries due to the large window sizes used for classification. Segmentation into a larger number of classes would accentuate the problem by increasing the number of region boundaries.

Our multiresolution segmentation algorithm utilizes the Weibull clutter model with a formulation similar to that found in [Krishnamachari and Chellappa, 1997]. The Weibull clutter model used for CFAR detection has proved reasonably accurate and has some characteristics similar to the K-distribution that has been found to be a good model for ADTS clutter [Yueh *et al.*, 1989]. The Weibull model has an advantage over the K model in that the log-likelihood function is simpler, resulting in faster implementation. Training is also more straightforward because an iterative ML algorithm for Weibull parameter estimation is available whereas a two-dimensional search through parameter space must be performed to find the ML estimate of the K parameters. Another advantage of the Weibull model is that it is appropriate for both magnitude and intensity images due to the fact that squaring a Weibull random variable produces another Weibull random variable.

Let X_s and L_s be the observed intensity and true label at the pixel location s, respectively. Under the Weibull clutter assumption, the conditional distribution of X_s given the label l can be expressed as

$$p(X_s|L_s = l) = \frac{C_l}{B_l}\left(\frac{X_s}{B_l}\right)^{C_l-1} \exp\left[-\left(\frac{X_s}{B_l}\right)^{C_l}\right]$$

where the parameter set (B_l, C_l) contains the shape and scale parameters computed for class l from the training set. A sliding window is used to form the joint log-likelihood function over a local neighborhood, for each class. It is assumed that all the pixels within the window have the same label. The pixel under test is given the class label which maximizes

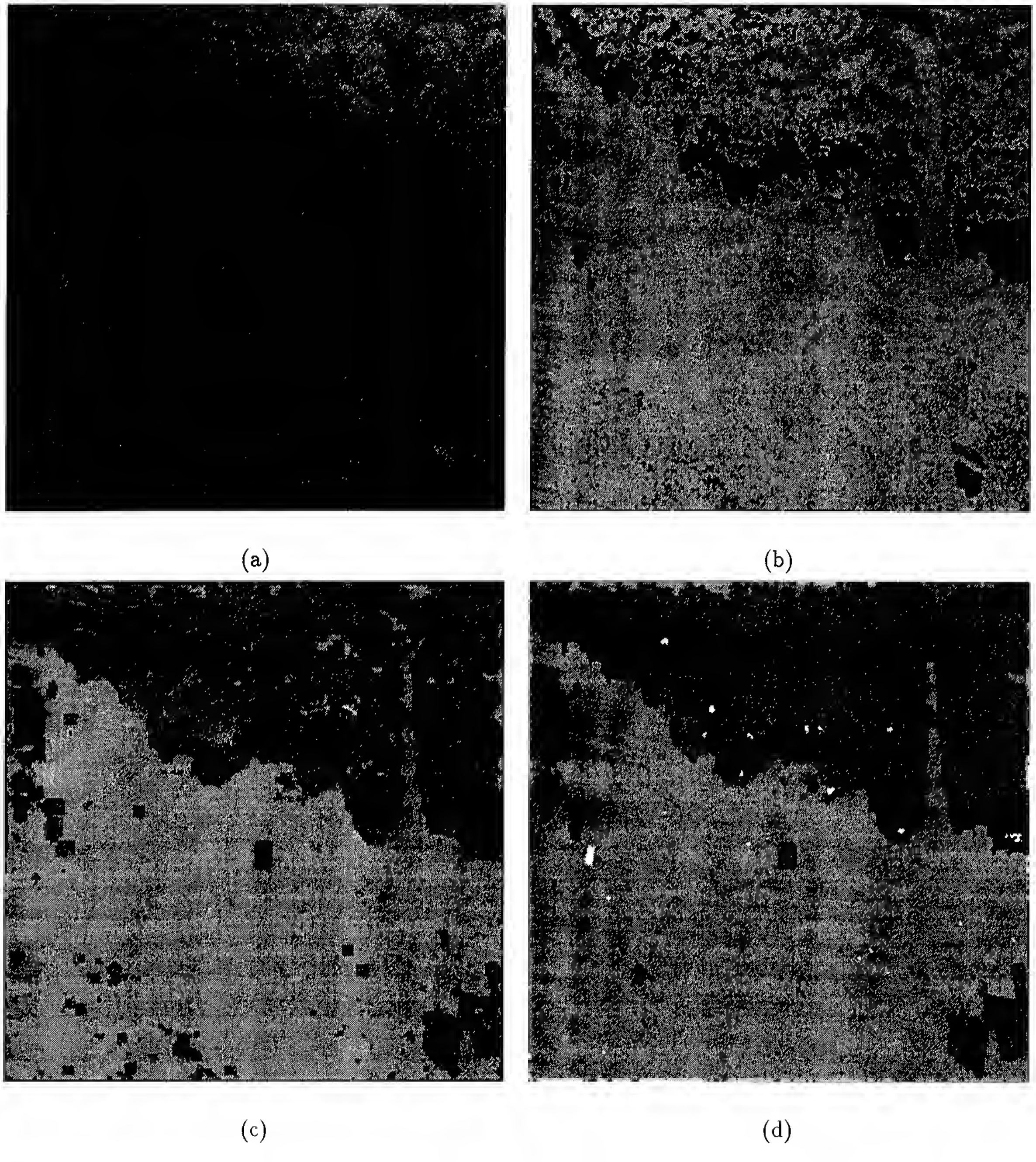


Figure 1: Results of multiresolution segmentation: (a) Original TESAR image, (b) single-resolution ML segmentation (light gray=grass, dark gray=trees, black=shadows), (c) multiresolution ML segmentation, (d) result of post-processing the multiresolution segmentation image with CFAR detections overlaid in white.

this joint log-likelihood function.

In the multiresolution formulation, we begin with a quadtree with M levels where $k \in \{0, 1, \ldots, M-1\}$ denotes a level of the quadtree. The lowest level, corresponding to $k = 0$, is the original fine-resolution $N \times N$ SAR intensity image. At level k, the image has been reduced to a coarser-resolution image of size $\frac{N}{2^k} \times \frac{N}{2^k}$.

In our experiments, we tried three different methods of resolution reduction—downsampling, half-band FIR filtering, and averaging the four child pixels. Somewhat surprisingly, we found that averaging the intensity values produced the best results. We have also found that a three-level tree works well, with little improvement obtained by adding levels.

Segmentation begins at the coarsest level of the quadtree. The resulting segmentation result is denoted by $L^{(k)}$ and is of size $\frac{N}{2^k} \times \frac{N}{2^k}$. Associated with $L^{(k)}$ is the confidence measure, $C^{(k)}$, defined as the difference between the log-likelihood value of the ML label and the log-likelihood of the runner-up label:

$$C_s^{(k)} = z_s^{ML} - z_s^{RU}$$

where z_s^{ML} and z_s^{RU} are the log-likelihoods associated with the ML label and the runner-up label for pixel s.

The segmentation at level $k-1$, $L^{(k-1)}$, is then initialized using a zero-order hold in each dimension to obtain

$$L_s^{(k-1)} = L^{(k)}_{\lfloor \frac{s}{2} \rfloor}$$

The confidence measures are propagated in the same way. At the new level, any pixel with confidence measure below the threshold $T^{(k)}$ will be relabeled based on estimates from the higher-resolution image. We expect that at boundary regions the coarse-resolution window will contain a mixture of regions and have low confidence. Segmenting at the higher resolution should produce a more confident labeling in boundary regions. This process continues until the finest scale is reached.

Figure 1 shows an example of TESAR imagery segmented into three classes (grass, trees and shadows) using the multiresolution segmentation scheme. Figures 1(b) and (c) contrast the results of ML segmentation at the finest resolution and the multiresolution segmentation. Finally, Figure 1(d) shows the multiresolution segmentation after some post-processing and reintroduction of the CFAR-detected targets. Post-processing involved removal of small regions and looking for supporting shadow evidence for tree regions.

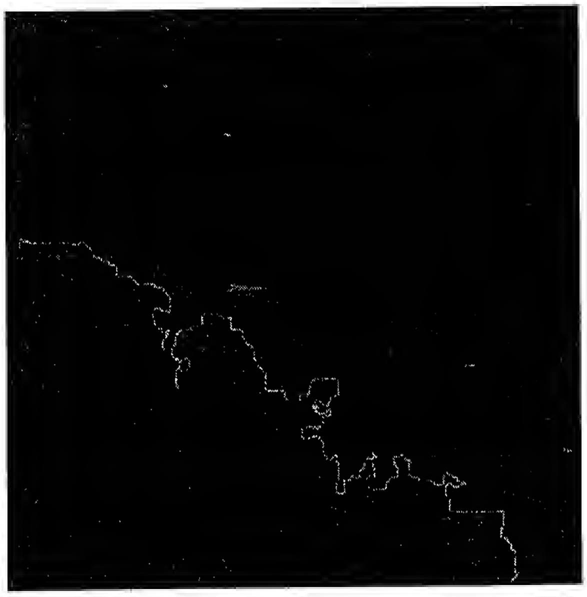


Figure 2: TESAR image with treeline extracted from grass/tree boundary after segmentation.

2.1 Treeline extraction and region-adaptive detection

One method of utilizing context from a single image is the extraction of treelines for use with region-adaptive target detection algorithms. In a region-adaptive detection algorithm, we run the CFAR detector with different thresholds in different regions. For non-foliage-penetrating radar, one could run CFAR with a low false alarm rate in the clear, use a different false alarm rate along region boundaries, and omit CFAR processing altogether in large homogeneous forest areas. Lowering the threshold in the boundary region may allow detection of targets in the presence of tree layover. Thus, extraction of the treeline facing the sensor becomes crucial. Partially occluded targets along the trailing treeline can usually be detected with a standard CFAR detector because of the low intensity of the shadow background, and adjusting the CFAR thresholds along this boundary could produce a large number of false alarms with no benefit obtained. Region adaptive detection and treeline extraction are applications where the smoothness provided by our multiresolution segmentation is important. Without a smooth segmentation, dominant treelines would be difficult to detect and the large number of small regions would render the region-adaptive algorithms ineffective.

Figure 2 shows an original TESAR image and a treeline extracted by the multiresolution segmentation algorithm. After extracting tree/grass boundary pixels, contours below a programmable size threshold were eliminated. We do not currently have imagery with good examples of layover, so our examples are limited to boundary extraction at this time.

3 Registration and Exploitation of Multipass SAR Imagery

Assuming a flat earth, a large range-to-swath-width ratio, and the availability of acquisition parameters, the registration of two images acquired from an airborne SAR can be approximated by an affine transformation [Kuttikkad *et al.*, 1997]. Thus, a 2-D point, $\boldsymbol{p}^{(1)}$, in the first image can be transformed to the corresponding point, $\boldsymbol{p}^{(2)}$, in the second image via the transformation

$$\boldsymbol{p}^{(2)} = \boldsymbol{A}\boldsymbol{p}^{(1)} + \boldsymbol{b} \tag{1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{\delta x^{(2)}} & 0 \\ 0 & \frac{1}{\delta r^{(2)}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \cos\theta^{(2)} \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\cos\theta^{(1)}} \end{bmatrix} \begin{bmatrix} \delta x^{(1)} & 0 \\ 0 & \delta r^{(1)} \end{bmatrix}$$

and $\boldsymbol{b}$ is a translation which can be determined from GPS information. Here $\delta x^{(i)}, \delta r^{(i)}$ are the respective pixel resolutions in the cross-range and range dimensions, $\theta^{(i)}$ are the depression angles, and $\phi = \phi^{(2)} - \phi^{(1)}$ is the difference in sensor headings between the two images. For convenience, the affine transformation of (1) can be written in the general matrix formulation

$$\begin{bmatrix} x^{(2)} \\ r^{(2)} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ r^{(1)} \\ 1 \end{bmatrix} \tag{2}$$

In order to compensate for errors in GPS-derived location, we extract a number of point features from each image and refine the translation parameters from them. The features chosen should lie in (or near) the ground plane, so that there are no layover effects that would affect different views differently. They should also be easy to detect and should persist across images. We have chosen the centroids of clusters of bright pixels as our point features. These bright returns result from metallic objects and other specular reflectors in the scene which may lie embedded in non-homogeneous background clutter. In the images we experimented with, they consist of stationary vehicles and other strong reflectors, like dihedrals and trihedrals. The Order Statistic CFAR [Rohling, 1983] technique is used to detect bright pixels in spatially-varying clutter. Terrain backscatter is modeled as a complex Gaussian resulting in a Rayleigh magnitude distribution. After initial registration, distances between each feature point in one image and all feature points in the other image are computed. A search is then performed to find the maximum number of one-to-one matches that result in the same approximate translation.

An example of registering multipass airborne SAR data is illustrated using three views of the Stockbridge target array in Figure 3. The alternate views were automatically registered with the reference image and transformed to its coordinate system, using our affine model (bottom).

3.1 Exploitation of registered imagery

Co-registered multi-pass imagery of an area can augment information about the scene and resolve ambiguities. We present some applications of registered multi-pass SAR imagery which would otherwise be difficult with a single image. Shadows in radar images indicate a lack of backscatter for a certain duration in the range gating window immediately behind a tall object. One obvious application of registered imagery is to fill in the missing information in shadow regions. The missing information in one image can be supplemented using the segmentation map from another co-registered image. Another use of registered multipass imagery is object height estimation which is considered next.

3.2 Estimation of object height

Topography reconstruction from a stereo pair of SAR images, acquired from the same side, opposite sides, or intersecting flight paths has been demonstrated ([Leberl, 1990], chs. 13,14). These techniques attempt to obtain a dense height map from two stereo images. Another technique for reconstructing terrain heights is to use radar interferometry, which requires two coherently acquired images [Zebker and Goldstein, 1986]. We consider the problem of acquiring heights of specific structures from a pair of non-interferometric SAR images collected from possibly intersecting flight paths. In the case of high-resolution airborne SAR imagery, the presence of speckle undermines any pixel-intensity-based matching technique and region-based techniques may not give a sufficiently dense height map. Moreover, registration errors may be on the order of a few pixels, leading to errors in pixel-by-pixel height computations. We therefore restrict our attention to detecting heights of man-made structures

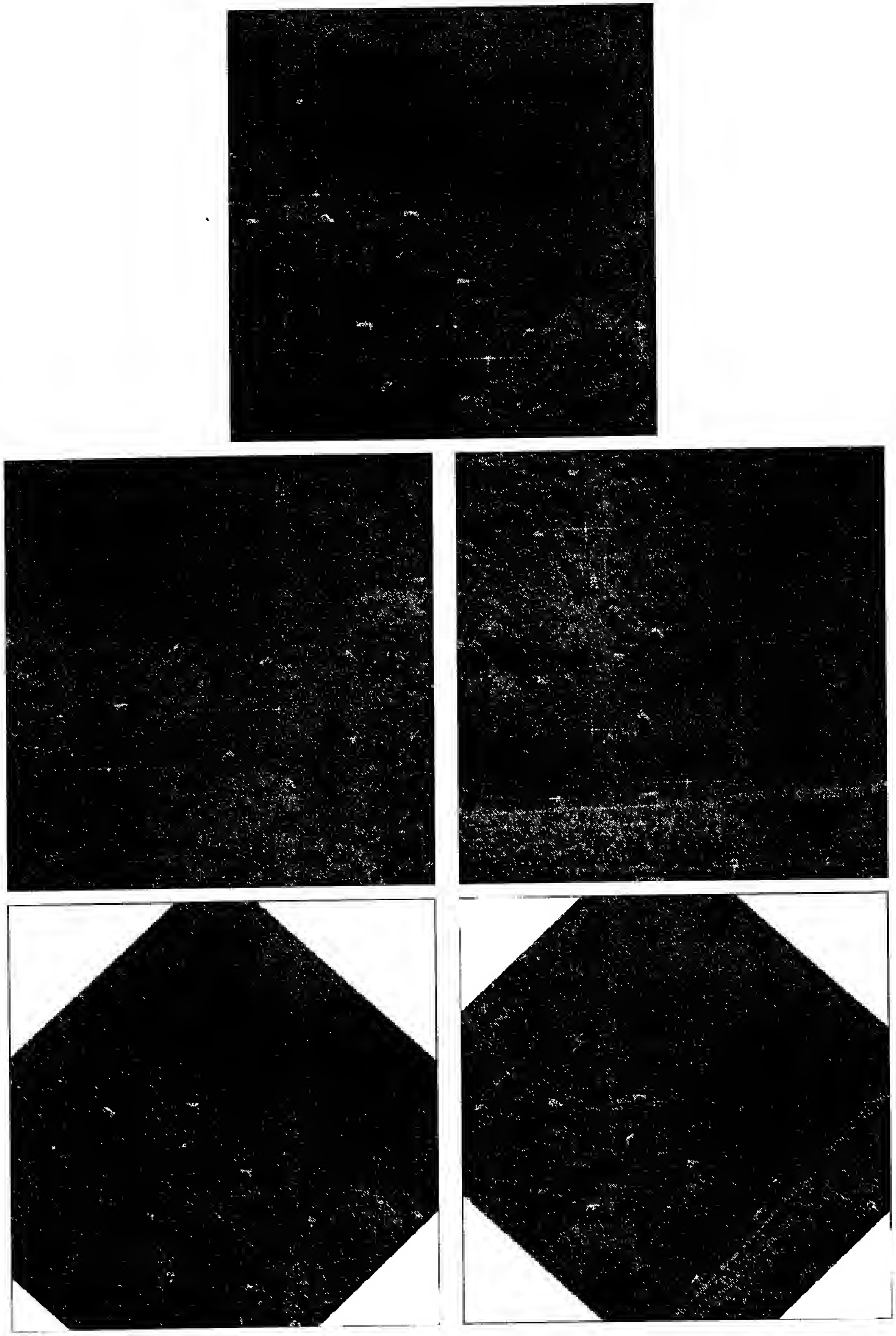


Figure 3: Registration example: Reference image (top), alternate views (middle), and other views registered to reference image (bottom).

like buildings and natural objects such as treetops, which are easier to detect and match.

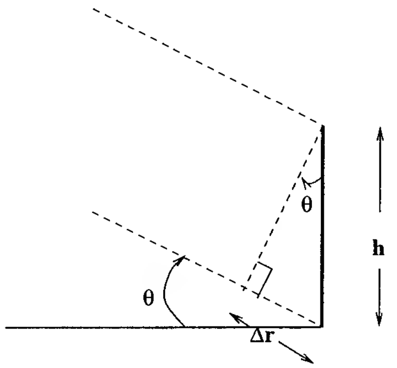


Figure 4: Observed range location of a tall object

Layover causes the slant plane image of the top of a structure of height h (Figure 4) to appear at range location

$$r_{obs}^{(i)} = r^{(i)} - \Delta r^{(i)} = r^{(i)} - h \sin\theta^{(i)}$$

The superscript i refers to image i, θ is the depression angle, and r and r_{obs} are the slant range to the base and top of the structure, respectively. Mathematically, it is possible to compute the height of a tall object, given the exact location of a single point on it in two views, and the transformation between the two images. In practice, the difficulty in automatically localizing point features in SAR images and inaccurate registration lead to erroneous height computation.

Linear features, which are easier to detect than point features, arise in SAR terrain imagery due to the cardinal streaks of buildings, treelines, road edges, vegetation boundaries, etc. We are specifically interested in extracting heights of buildings and trees. Due to layover, the tops of the vertical sides of buildings and treelines, closest to the radar, are at nearer ranges than their bases. These features can be automatically detected using segmentation/labeling or can be manually selected from the image pairs.

Let $\boldsymbol{l}^{(i)}$ be the location of the base of a linear structure of height h in image i. In a digital image, a line $\boldsymbol{l}$ can be thought of as a collection of points $\boldsymbol{p}_j$:

$$\boldsymbol{l}^{(i)} = \{\boldsymbol{p}_j^{(i)}\} = \{[\ x_j^{(i)} \quad r_j^{(i)} \quad 1\ ]^T\}$$

Since layover affects the range location of the top of a vertical structure, its apparent location is given by the collection of points

$$\tilde{\boldsymbol{p}}_j^{(i)} = \boldsymbol{p}_j^{(i)} - h \sin\theta^{(i)} * [\ 0 \quad 1 \quad 0\ ]^T$$

Let $\hat{\boldsymbol{l}}^{(1)}(=\{\hat{\boldsymbol{p}}_j^{(1)}\})$ be the affine projection (according to (2)) of $\tilde{\boldsymbol{l}}^{(1)}$ in the second image:

$$\hat{\boldsymbol{p}}_j^{(1)} = \boldsymbol{A}_3\tilde{\boldsymbol{p}}_j^{(1)} = \boldsymbol{A}_3(\boldsymbol{p}_j^{(1)} - h\sin\theta^{(1)} * [0\ 1\ 0]^T) \quad (3)$$

where $\boldsymbol{A}_3$ is the (3×3) affine transformation matrix of (2). Given that the locations of points along the base of the structure in the two images are related by $\boldsymbol{p}_j^{(2)} = \boldsymbol{A}_3 * \boldsymbol{p}_j^{(1)}$, (3) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{p}}_j^{(1)} &= \boldsymbol{p}_j^{(2)} - h\sin\theta^{(1)}\boldsymbol{A}_3 * [\ 0 \quad 1 \quad 0\ ]^T \qquad (4) \\ &= \tilde{\boldsymbol{p}}_j^{(2)} + h(\sin\theta^{(2)}\boldsymbol{I}_3 - \sin\theta^{(1)}\boldsymbol{A}_3) * [0\ 1\ 0]^T \end{aligned}$$

where $\boldsymbol{I}_3$ is the (3×3) identity matrix. Substituting for $\boldsymbol{A}_3$, (4) becomes

$$\hat{\boldsymbol{p}}_j^{(1)} - \tilde{\boldsymbol{p}}_j^{(2)} = \begin{bmatrix} -a_{12}h\sin\theta^{(1)} \\ h(\sin\theta^{(2)} - a_{22}\sin\theta^{(1)}) \\ 0 \end{bmatrix}$$

Although it is not possible to detect pairs of corresponding points on the two lines, it is possible to compute height, h, by computing the perpendicular distance, ρ, between them:

$$h = \frac{\rho}{(\sin\theta^{(2)} - a_{22}\sin\theta^{(1)})\sin\alpha - a_{12}\sin\theta^{(1)}\cos\alpha}$$

where α is the angle made by a line perpendicular to either of $\hat{\boldsymbol{l}}^{(1)}$ or $\tilde{\boldsymbol{l}}^{(2)}$ with the positive cross-range axis in the second image.

In practice, it is difficult to ensure that the projected line $\hat{\boldsymbol{l}}^{(1)}$ is exactly parallel to the observed line $\tilde{\boldsymbol{l}}^{(2)}$. Since the above technique for height extraction relies on the distance between two parallel line segments and their slope, refinements have to be made to the original line segment locations, to make them parallel. In order to achieve this, the observed lines in the two images are de-rotated by equal and opposite amounts about their midpoints. This is justified, since the line extraction technique is the same in both images, and any errors are expected to affect both images in the same statistical sense. It is not difficult to show that the corresponding rotation angle, φ, is the solution of

$$\frac{a_{21} + a_{22}\bar{m}^{(1)}}{a_{11} + a_{12}\bar{m}^{(1)}} = \frac{-\sin\varphi + \tilde{m}^{(2)}\cos\varphi}{\cos\varphi + \tilde{m}^{(2)}\sin\varphi}$$

where $\tilde{m}^{(i)}$ is the slope of the observed line in image i, and

$$\bar{m}^{(1)} = \frac{\sin\varphi + \tilde{m}^{(1)}\cos\varphi}{\cos\varphi - \tilde{m}^{(1)}\sin\varphi}$$

4 False Alarm Reduction Using Context

In this section, we demonstrate the use of context, derived from single or multipass imagery, for false alarm reduction. In the images we experimented with, stationary vehicles, cardinal streaks from buildings, and other strong reflectors, like dihedrals and trihedrals, produced bright returns. If only a single image of a scene is available, a site model can provide context for ATR algorithms, by identifying regions where targets may be expected to be present. In the case of data collected from multiple passes over the same site, the images can be registered and used for providing consistency-based false alarm reduction.

4.1 Context from a single image

In Section 2, we considered schemes for delineating natural regions, such as forests and lakes, as well as identifying man-made objects, such as buildings and roads. Since military ground vehicles are not expected in dense forest or water, bright pixel clusters surrounded by these labels can also be removed as false alarms.

Figure 5 shows the SAR image of an urban area. CFAR processing of the original image produced more than 1500 candidate target clusters in this complex scene. Inspection proved less than 10% of these clusters to be cars. The remaining are false alarms due to buildings, railroads, bridge railings, etc. We use the site model from the backscatter image to create a mask for buildings. False alarm mitigation begins with the removal of streaks from buildings using this mask. Discrimination can then be performed on the remaining target chips using a clutter training set.

An example obtained from TESAR imagery is shown in the top image of Figure 6. The middle image is the segmented image with detection results overlaid. In addition to several false alarms in foliage, there are two partially occluded vehicles, two vehicles in the clear, and several trihedrals which we consider to be targets. Based on the segmentation results, we can declare detections in heavy clutter to be unreliable; these are marked by boxes in the bottom image. After extracting the context by finding the proportion of the different labels in a hollow window surrounding each detection, two of the false alarms can be eliminated because they are closely surrounded by trees and are therefore unlikely to be ground vehicles. The other false alarms and the two occluded targets can be flagged as obstructed, but not eliminated, because they are close enough to the treeline that we cannot distinguish isolated trees from targets.

4.2 Context from multipass imagery

Registered SAR imagery can be used for reducing the number of candidate targets. Bright pixels detected by CFAR processing are grouped into clusters and small clusters are eliminated. False alarms due to buildings and in foliage can be eliminated using the technique described earlier. The targets may be in the open, partly occluded by shadow or the layover of trees, or camouflaged. Our claim is that false alarms and other directional reflectors (e.g. dihedrals) will not be consistently visible in multiple views, whereas complex targets can be observed in all views. Therefore we look for consistent bright pixel clusters in multiple views to reduce the false alarms. We take care of occlusions by shadows or trees by incorporating them in our consistency check using the segmentation map of the scene. We do not address the camouflage issue in this paper.

A second, registered view of Figure 6 is shown in Figure 7 (top image) and the registered segmentation and CFAR results are shown in the middle image. Using the new CFAR detections, we can verify that the two false alarms eliminated using only the first pass were truly false alarms and four of the other five false alarms can also be eliminated. With the context information provided by segmentation, these false alarms are eliminated because corresponding detections in the second pass do not exist and the detection points are labeled as trees or clear. The remaining false alarm cannot be removed because the detection point in the second image is in shadow. We can also verify that the two partially occluded targets are indeed targets and not isolated trees because they are detected in clear areas of the second-pass image. The bottom image of Figure 7 shows the result with all but one false alarm eliminated.

Figure 8 shows examples of multipass Stockbridge imagery, which demonstrate false alarm reduction techniques using multiple views. The top image shows a reference view with CFAR-detected pixel clusters marked. Real and calibration targets (marked by boxes) as well as false alarms (marked by ovals) were manually identified. The middle row shows two other views similarly processed. Notice that some calibration targets disappear while others appear. Some of the real targets are either partially or completely occluded by shadows and trees. A lot of false alarms show up in foliage where clumps of the canopy are surrounded by shadow regions. The reference image is registered with, and transformed to, the coordinate system of each of the two new images, using the method described in Section 3. Consistent target clusters are marked as targets in the bottom row, while others are removed. In the case of areas with no overlap after registration, the

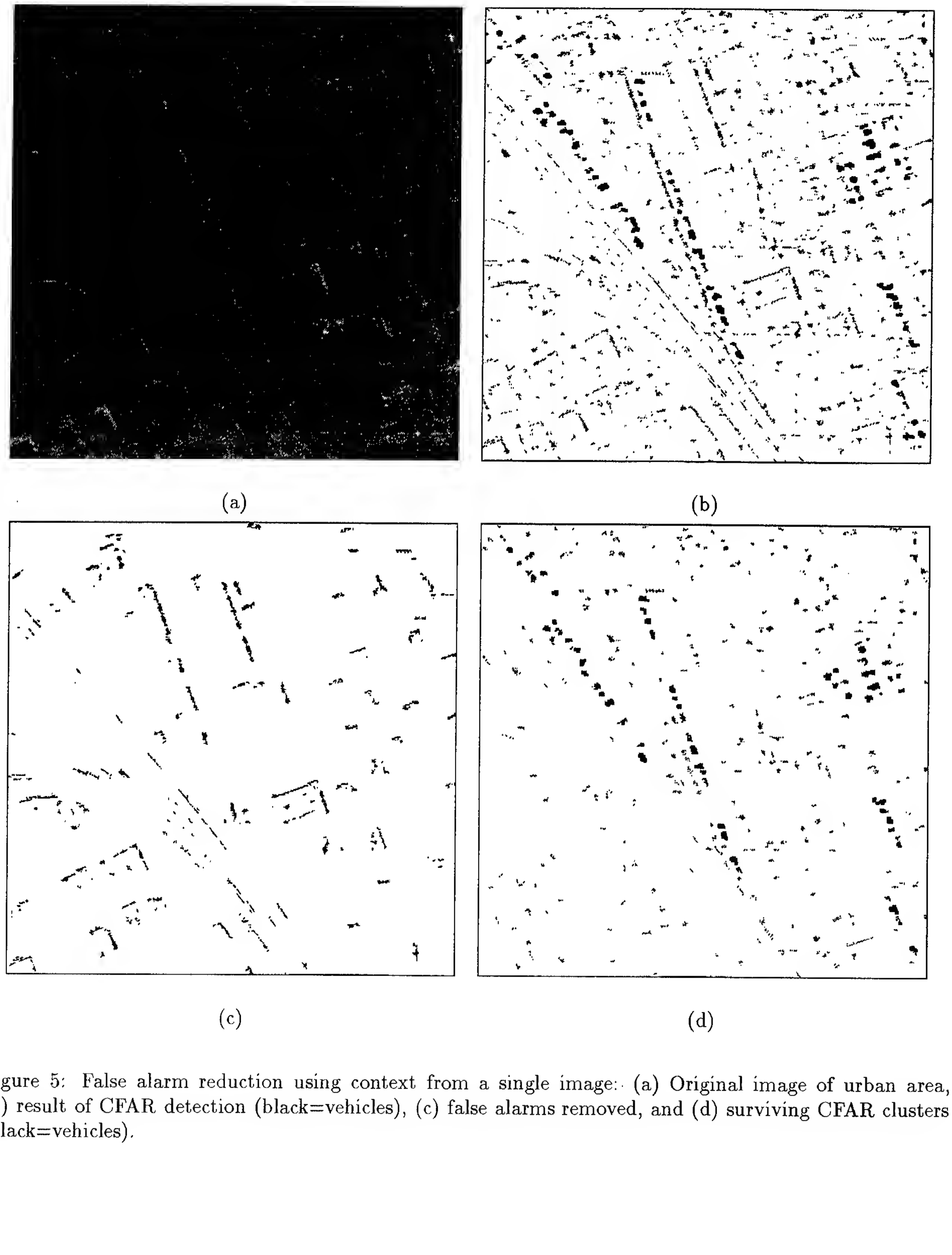


Figure 5: False alarm reduction using context from a single image: (a) Original image of urban area, (b) result of CFAR detection (black=vehicles), (c) false alarms removed, and (d) surviving CFAR clusters (black=vehicles).

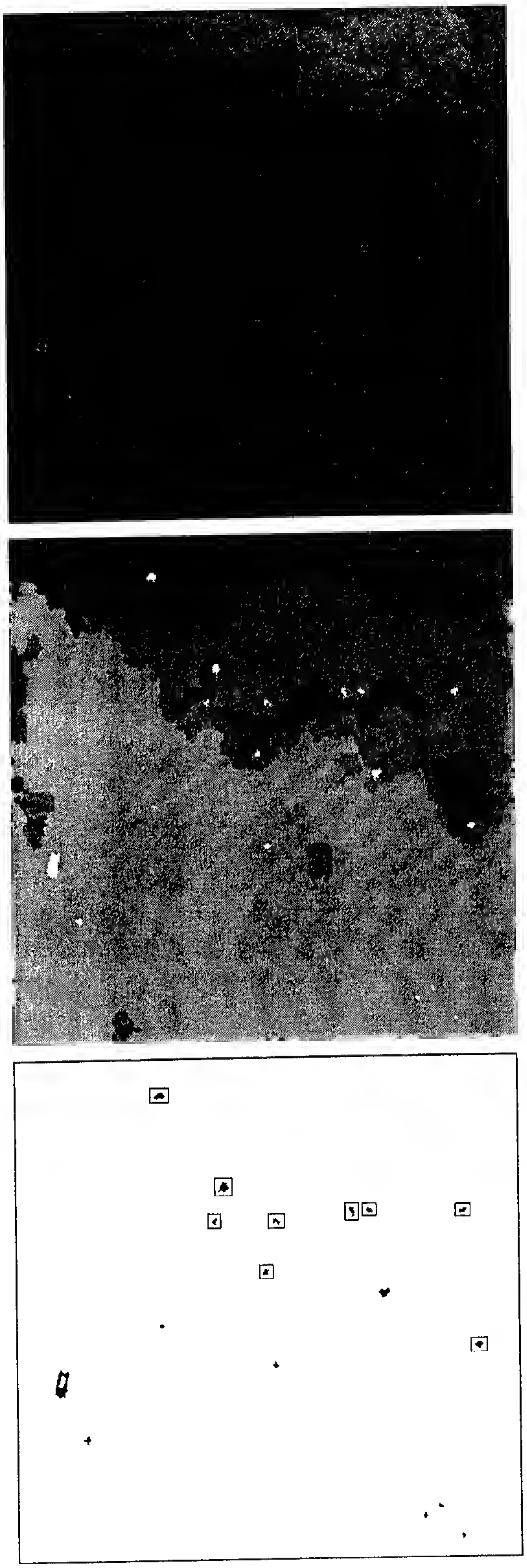


Figure 6: False alarm reduction using context derived from a single TESAR image: Original image, result of two-pass Weibull OS CFAR, and unreliable detections (marked with boxes) based on multiresolution segmentation.

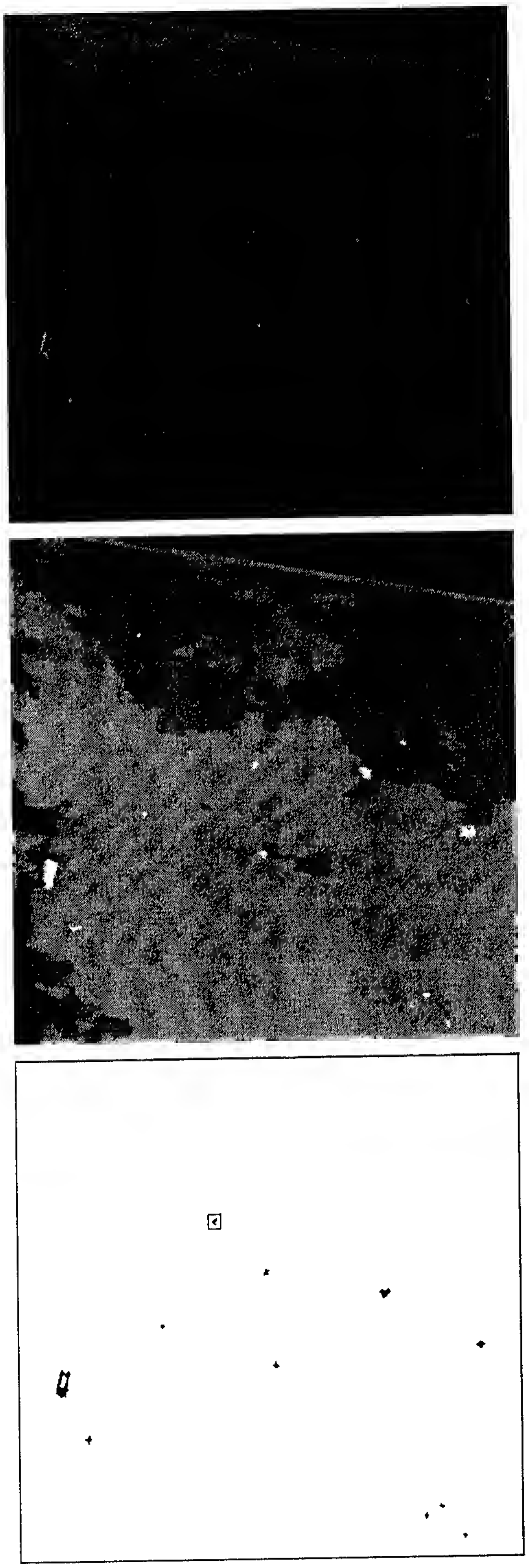


Figure 7: False alarm reduction using registered TESAR imagery: Second image, CFAR results on the new image, and confirmed detections after combining the evidence from both views.

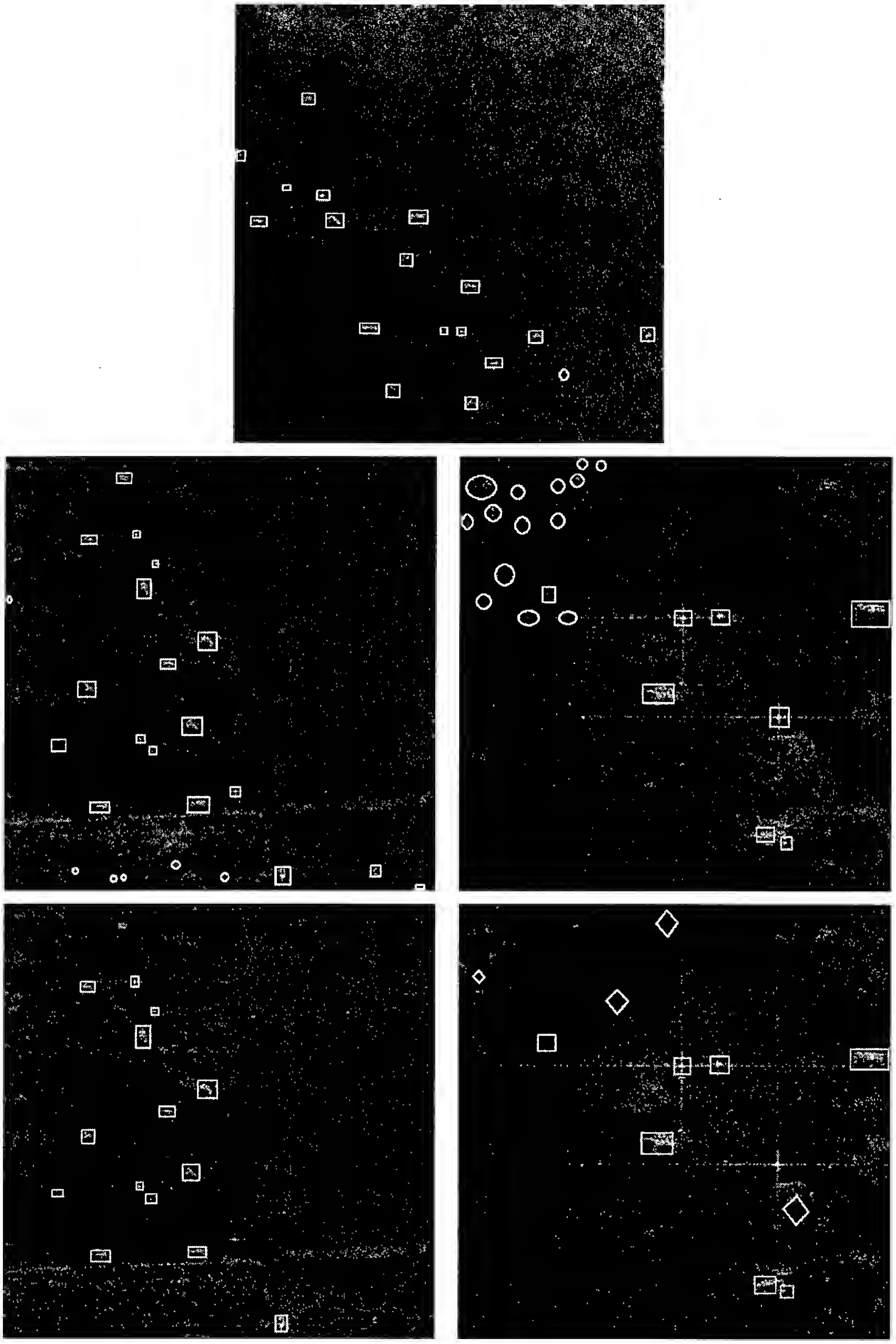


Figure 8: False alarm reduction using context derived from multiple images. Top: Ground truth image (boxes=targets, ovals=false alarms). Middle: Alternate views with manually classified targets. Bottom: Automatically classified targets using context from ground truth image (black boxes=no data, diamonds=possible new targets).

corresponding clusters are marked by black boxes. The most interesting result shows up in the bottom-right image. Here, some pixels are marked by diamonds. These were pixel clusters in the reference image which overlapped with regions of shadow or tree in the test image. An example of the former is the diamond in the top center, and of the latter, the diamond in the bottom right. The first represents a target completely surrounded by shadow as indicated by the segmentation of the test image, and the second is a target surrounded by a tree region. In reality, these were targets which were, respectively, in shadow and laid over by a tree in the test image.

5 Discrimination

We first remove clusters of CFAR-detected pixels which are determined to be false alarms, either from the context derived from a single image or because of lack of consistency in multi-pass imagery. Candidate target chips are generated by grouping the surviving pixels and computing their bounding boxes. A set of features is computed for each target chip and compared to a feature set derived from the clutter training set.

We use a combination of the features developed at Lincoln Laboratory [Kreithen *et al.*, 1993; Novak *et al.*, 1993] for single-polarization data, namely, standard deviation, weighted rank fill ratio and fractal dimension, and features related to the size of the target chip, namely, area, length and width. These features are summarized in Table 1. P_{ij} is the pixel at the (i,j)th image location, and $P_{ij}^{(k)}$ are those pixels that belong to the target chip $C^{(k)}$. N is the total number of pixels in the target chip, and X and Y are the dimensions (in image coordinates) of the upright bounding box of the region $(X \geq Y)$.

If templates of targets are available we use a target training set and compare features of templates to those of target chips. If not, since the set of possible types of targets is quite large, we compare the features extracted from the candidate target chips to those derived from clutter only. As many clutter training sets are built as there are types of clutter observed in a standard training image. A Mahalanobis-like distance measure is computed for each chip:

$$Z_i = \frac{1}{n} \times (X_i - M_{tr})^T \times S_{tr}^{-1} \times (X_i - M_{tr}) \quad (5)$$

where n is the number of features used. This distance is compared to a trained threshold to discriminate non-clutter objects from clutter.

Examples of discrimination applied to two images from Lincoln Lab's Stockbridge Target Array, consisting of 7 to 8 targets, are shown in Figure 9. CFAR processing produced approximately 20 candidate target chips. We used Xpatch-generated templates of two targets (M48 and M113) to compute the target features. All features of candidate target chips are compared to those of the training set using the distance measure of (5). The threshold for discriminating targets from clutter was trained using one view and applied to the others. Figure 9 shows the results of reduction of false alarms. Three sets of discrimination features were tested—Lincoln Laboratory features only (standard deviation, weighted rank fill ratio, fractal dimension), fractal dimension and size features (area, length, width), and a combination of all the features. Table 2 shows the results of discrimination applied to three passes of the Stockbridge Target Array. The distance threshold for discrimination was set to retain all the targets. Performance can be judged by looking at the number of false alarms in the output of the discriminator. We found that Lincoln Laboratory's single-polarization features alone were not sufficient for good discrimination. The least false alarms were produced by the size and shape features, which seem to work best at this resolution.

6 Conclusion

Use of context can greatly reduce the burden on ATR algorithms by reducing the number of candidate targets. We define context to be a region delineation for a single image and detection consistency across registered imagery for a multipass scenario. For the single-pass case, context is derived using intensity data and a multiresolution segmentation scheme. We have also formulated the registration equation for images of the same site acquired from an airborne SAR platform. Examples of context-derived false alarm reduction for both cases were presented. Finally, a set of shape- and texture-based features was used for discriminating targets from clutter. Experimental results using TESAR and ADTS data were presented.

7 Acknowledgments

We wish to thank Don Wineberg and Vince Zagardo of Northrop Grumman Corporation for obtaining permission to publish the TESAR imagery. We also wish to thank Les Novak of MIT Lincoln Laboratory for providing the ADTS data.

Features	Expression	Comments
Standard Deviation	$\sigma = \sqrt{\frac{(S_2 - S_1^2)/N}{N-1}}$	$S_i = \sum_{C^{(k)}} [10 \log_{10} P_{ij}]^i$for $i = 1, 2$.
Weighted Rank Fill Ratio	$\nu = \frac{\sum_R l \text{ brightest } P_{ij}^{(k)}}{\sum_R P_{ij}}$	$R = (100 \times 100)$ square $l = 5\%$ of size-of(R)
Fractal Dimension	$FD = \log_2(M_1/M_2)$	M_i = No. of boxes of size i reqd. to fill target cluster
Area	$\sum_{C^{(k)}} P_{ij}^{(k)}$	
Length	$L = \sqrt{X^2 + Y^2}$ $= \max(X, Y)$	If orientation = 45°, 135° If orientation = 0°, 90°, or none
Width	$W = \frac{Y}{X/(\sqrt{X^2+Y^2})}$	

Table 1: Feature set for discrimination

Features	Pass 1 (18 detections)		Pass 2 (16 detections)		Pass 3 (25 detections)	
	Targets	False Alarms	Targets	False Alarms	Targets	False Alarms
Lincoln Laboratory	16	4	13	4	15	5
Size and Shape	13	1	10	1	10	0
All Together	15	3	11	2	14	4

Table 2: Results of the discrimination for the Stockbridge target array data

References

[Chellappa *et al.*, 1996] R. Chellappa, S. Kuttikkad, R. Meth, P. Burlina, K. Eom, and C. Shekhar. Model-supported exploitation of SAR imagery. *Proceedings, DARPA Image Understanding Workshop*, pp. 389–407, Palm Springs, CA, February 1996.

[Fosgate *et al.*, 1997] C. Fosgate, H. Krim, W. Irving, W. Karl, and A. Willsky. Multiscale segmentation and anomaly enhancement of SAR imagery. *IEEE Trans. on Image Processing*, 6:7–21, January 1997.

[Kreithen *et al.*, 1993] D.E. Kreithen, S.D. Halversen, and G.J. Owirka. Discriminating targets from clutter. *Lincoln Laboratory Journal*, 6(1):25–51, 1993.

[Krishnamachari and Chellappa, 1997] S. Krishnamachari and R. Chellappa. Multiresolution Gauss-Markov random field models for texture segmentation. *IEEE Trans. on Image Processing*, 6:251–267, February 1997.

[Kuttikkad and Chellappa, 1995] S. Kuttikkad and R. Chellappa. Building wide-area 2-D site models from high resolution fully polarimetric Synthetic Aperture Radar images. *Proceedings, IEEE Intl. Symposium on Computer Vision*, pp. 389–394, Coral Gables, FL, November 1995.

[Kuttikkad *et al.*, 1996] S. Kuttikkad, R. Chellappa, and L. M. Novak. Building wide-area 2-D site models from single- and multipass single-polarization SAR data. In *Proc. SPIE: Algorithms for Synthetic Aperture Radar Imagery 3*, pp. 34–44, Orlando, FL, April 1996.

[Kuttikkad *et al.*, 1997] S. Kuttikkad, R. Meth, and R. Chellappa. Registration and exploitation of multi-pass airborne Synthetic Aperture Radar images. CfAR Technical Report, to appear.

[Leberl, 1990] F. W. Leberl. *Radargrammetric Image Processing.* Artech House, Inc., Norwood, MA, 1990.

[Novak *et al.*, 1993] L.M. Novak, G.J. Owirka, and C.M. Netishen. Performance of a high-resolution polarimetric SAR automatic target recognition system. *Lincoln Laboratory Journal*, 6(1):11–24, 1993.

[Rohling, 1983] H. Rohling. Radar CFAR thresholding in clutter and multiple target situations. *IEEE Trans. on Aerospace and Electronic Systems*, 19:608–621, July 1983.

[Yueh *et al.*, 1989] S. H. Yueh, J. A. Kong, J. K. Jao, R. T. Shin, and L. M. Novak. K-distribution and polarimetric terrain radar clutter. *Journal of Electromagnetic Waves and Applications*, 3:747–768, August 1989.

[Zebker and Goldstein, 1986] H. Zebker and R. Goldstein. Topographic mapping from interferometric SAR observations. *Journal of Geophysical Research*, 91(B5):4993–4999, 1986.

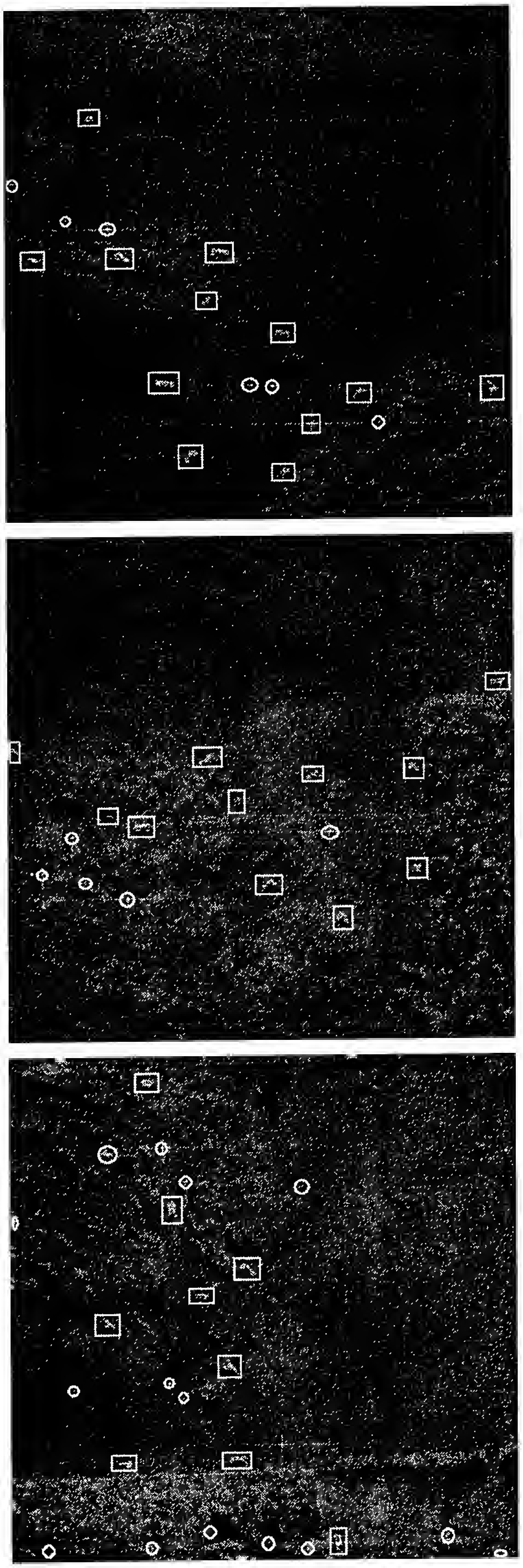


Figure 9. Discrimination example; boxes are targets and ovals are false alarms.

A Parametric Model for Synthetic Aperture Radar Measurements

Mike Gerry, Lee Potter, and Randy Moses*
Department of Electrical Engineering
The Ohio State University
Columbus, Ohio 43210
E-MAIL: {gerry.2,potter.36,moses.2}@osu.edu
HOMEPAGE: http://er4www.eng.ohio-state.edu/ips/

Abstract

We present a parametric model for radar scattering as a function of frequency and aspect angle. The model is used for analysis of synthetic aperture radar data. The estimated parameters provide a concise, physically relevant description of measured scattering for use in target recognition, data compression and scattering studies. The scattering model and an image domain estimation algorithm are applied to two measured data examples.

1 Introduction

At high frequencies, the scattering response of an object is well approximated as a sum of responses from individual scattering centers [Keller, 1962]. These scatterers provide a physically relevant, yet concise, description of the object and are thus good candidates for use in target recognition, radar data compression, and scattering phenomenology. In this paper we consider the analysis of radar data measured as a function of frequency and aspect angle. We develop a parametric scattering model for this two-dimensional problem. The model is based on both the physical optics and the geometric theory of diffraction (GTD) monostatic scattering solutions and extends the one-dimensional GTD model presented in [Potter, et al., 1995] to include aspect angle. Our model provides a physical description of target scattering centers, each of which is described by a set of parameters describing position, shape, orientation (pose) and relative amplitude. This is a richer description of target scattering than is available either from conventional Fourier-based imaging techniques [Mensa, 1991] or from less physically accurate point scattering parametric models [Tu, et al., 1997].

Recent developments in mechanism extraction from two-dimensional radar data [Tu, et al., 1997], [Sacchini, et al., 1993] are based on the assumption that scattering centers are localized to isolated points. While this description is valid for many scattering centers at many aspect angles, some common scattering mechanisms behave as distributed elements, and point scattering models fail to accurately model the scattering. The aspect dependence in our two-dimensional model allows description of both localized and distributed scattering centers, providing a higher fidelity description of scattered fields. The improved model provides the potential both for improved data compression and for the discrimination of localized versus distributed scattering mechanisms.

We develop a simple parametric model of far-field scattering as a function of frequency and aspect angle. The parameters are treated as unknown, deterministic quantities. We present an

*This work was sponsored by the Defense Advanced Research Projects Agency under contract F33615-97-1020 monitored by Wright Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.

algorithm to estimate the parameters from an image domain representation of the measured data. Estimation in the image domain, rather than in the frequency–aspect domain, provides the advantages of clutter suppression, model order reduction, and computational savings.

The paper is organized as follows. In Section 2 we present the two-dimensional scattering model based on GTD and physical optics solutions for simple geometries. In Section 3 we transform the frequency–aspect angle domain model into the image domain for the purpose of parameter estimation. In Section 4 we present an algorithm for estimation of the unknown parameters of the model from measured data. In Section 5 we present the Cramér-Rao lower bound (CRB) on the variance of the estimated model parameters, and discuss practical implications of the CRB for parameter uncertainty. In Section 6 we present experimental results obtained by applying our estimation algorithm to data measured in a compact-range anechoic chamber.

2 Model Development

We develop a parametric model for the backscatter from objects measured as a function of frequency and aspect angle. We seek a model that maintains high fidelity to the scattering physics for many objects, yet is sufficiently simple in its functional form to permit robust inference from estimated parameters.

For this development, we assume a data collection scenario consistent with synthetic aperture radar (SAR) imaging. A reference point is defined, and we require that the radar trajectory and reference point are co-planar. We label this imaging plane using an $x-y$ Cartesian coordinate system with origin at the reference point. The radar position is then described by an angle ϕ defined counterclockwise from the x direction. We assume far zone backscatter, and therefore obtain plane-wave incidence on objects.

From the geometric theory of diffraction (GTD) [Keller, 1962] and its uniform version [Kouyoumjian and Pathak, 1974], if the wavelength of the incident excitation is small relative to the target extent, then the backscattered field from an object consists of contributions from electrically isolated scattering centers. In developing our model, we proceed in a similar fashion and characterize the frequency and aspect angle dependence of individual scattering centers. Each scattering center is described by a small number of parameters. The total scattered field from a target is then modeled as the sum of these individual scatterers.

We make three assumptions about the far zone backscattered field, and each assumption leads to the functional form for a portion of our scattering model. First, phase dependence is linear and defined by the position of the scattering center. Second, amplitude dependence on frequency is defined by the high-frequency approximation derived from the GTD. Third, amplitude dependence on aspect angle is defined by characterizing the scattering center as either spatially localized or distributed. We consider these three dependencies, each in turn, to arrive at a parametric scattering model.

First, we consider only far-field scattering with a linear phase dependence with frequency. The phase of a scattering center, at a given aspect angle, is determined by the down range position of the scatterer. Accordingly, the backscattered field of the nth scattering center is expressed

$$E_n^s(k,\phi) = S_n(k,\phi)\exp\{-j2k\hat{r}\cdot\vec{r}_n\} \qquad (1)$$

where $k = 2\pi f/c$ is the wave number, f is frequency in Hertz, c is the propagation velocity, ϕ is the aspect angle, $\hat{r}$ is the unit vector in the direction of the scattered field, and $\vec{r}_n = [x_n, y_n]$ is the position vector of the nth scattering center projected to the plane. The $e^{-j2\pi ft}$ time convention is assumed and suppressed. Here we consider only the co-polarized field; as such, all field quantities are written as scalars. The development is easily extensible to multiple polarizations [Chiang, 1996]. In summary, the phase dependence of our model describes the location of each scattering center in the plane of the radar measurement.

Second, we consider the amplitude dependence on frequency. In presenting the GTD, Keller [Keller, 1962] uses a conservation of energy ar-

Table 1: Alpha values for canonical scatterers.

α	Example scattering geometries
1	flat plate at broadside; dihedral
$\frac{1}{2}$	singly curved surface reflection
0	point; sphere; straight edge specular
$-\frac{1}{2}$	edge diffraction
-1	corner diffraction

gument to propose that the field diffracted from a point on an edge is proportional to $(jk)^{-\frac{1}{2}}$, and the field diffracted from a vertex is proportional to $(jk)^{-1}$. The simplicity of the GTD is that many practical object geometries give rise to a sum of these two scattering mechanisms. In Plonus, et al., 1978] and [Ross, 1966], it is shown that in addition to the edge and vertex diffraction, a larger class of scattering geometries also fit the $(jk)^{\alpha}$ power dependence on frequency, where the parameter α has a half integer value (see Table 1).

Third, we consider aspect dependence of scattering amplitude. As aspect angle as varied, we assume that a scattering center behaves in one of two ways: either a scatterer is localized and appears to exist at a single point in space, or it is distributed in the imaging plane and appears as a finite, nonzero-length current distribution. The amplitude dependence on aspect angle is different for each of these scenarios, and we seek a model that accounts for both scattering behaviors in a physically accurate, yet simple, functional form.

Examples of point mechanisms are trihedral reflection, corner diffraction, and edge diffraction. All of these mechanisms have slowly varying amplitude as a function of aspect angle. We exploit the commonality of point mechanisms by modeling this slowly varying function with a damped exponential

$$S_n(f,\phi) = A_n \exp(-2\pi f \gamma_n \sin\phi) \qquad (2)$$

The exponential function provides a mathematically convenient approximation containing only a single parameter. Although physical insight is used to arrive at the exponential model, the parameter γ_n has no direct physical interpretation.

On the other hand, examples of distributed scattering mechanisms are flat plate reflection, dihedral reflection, and cylinder reflection. Each of these scattering mechanisms has an amplitude dependence on aspect angle that contains a sinc(x) = $\frac{\sin(x)}{x}$ function. In all cases this sinc(x) function is the dominant term in the physical optics far-zone scattering solution, and we adopt the sinc(x) function to characterize angle dependence in the scattering model for scattering centers that are distributed:

$$S_n(f,\phi) = A_n \mathrm{sinc}\left(kL_n \sin(\phi - \phi_n)\right) \qquad (3)$$

where L_n is the length and ϕ_n is the orientation angle of the distributed scatterer.

We combine the different model terms from the point and the distributed scattering mechanisms to write our 2-D scattering model in a single expression

$$\begin{aligned} E_n^s(f,\phi) &= A_n \left(j\frac{f}{f_c}\right)^{\alpha_n} \\ &\quad \cdot \mathrm{sinc}\left(\frac{2\pi f}{c} L_n \sin(\phi - \phi_n)\right) \\ &\quad \cdot \exp(-2\pi f \gamma_n \sin\phi) \\ &\quad \cdot \exp(-j\frac{4\pi f}{c}(x_n \cos\phi + y_n \sin\phi)) \end{aligned} \qquad (4)$$

where $L_n = 0$ if the scattering center is localized, and $\gamma_n = 0$ if the scatterer is distributed. The parameter A_n is a relative amplitude for each scattering center. The total scattered field is a sum of p individual scattering terms.

$$E^s(f,\phi) = \sum_{n=1}^{p} E_n^s(f,\phi) \qquad (5)$$

The scattering model in Eqn. 5 is a function of frequency and aspect angle and is described by the parameter set $(x_n, y_n, \alpha_n, \gamma_n, L_n, \phi_n)$ for $n = 1, ..., p$. The parameters provide a rich physical description of the scatterers that are present in the data set. Each parameter, with the exception of γ_n, has a direct physical interpretation. Example scattering geometries distinguishable by their (α, L) parameters are presented in Table 2. The model is based on scattering physics and is developed to describe a

Table 2: Parameters α and L serve to discriminate many scattering geometries.

Example scattering geometries	α	L
dihedral	1	$L \neq 0$
corner reflector	1	0
cylinder	$\frac{1}{2}$	$L \neq 0$
sphere	0	0
edge diffraction	0	$L \neq 0$
corner diffraction	-1	0
double corner diffraction	-2	0

large class of scatterers while still maintaining a relatively simple form.

3 Transformation of Model into Image Domain

Measured radar data collected as a function of frequency and aspect is commonly processed coherently to form an image for display and interpretation. The magnitude or envelope of the complex-valued image provides an intuitive picture of the scattering behavior of the target.

The image domain provides several advantages for estimation of the unknown parameters in Eqn. 4. The four main advantages are: (1) clutter suppression, (2) reduction in local model order, (3) reduction in computation cost, and (4) insertion into multi-stage SAR target detection processing. These advantages are a result of the fact that radar imagery provides a temporal decomposition of the measured data. In the following paragraphs, we consider each of these advantages.

First, in the image domain we apply the model to high energy regions in order to accomplish clutter suppression. There exist many image segmentation algorithms [Stach and LeBaron, 1996] to automatically parse a radar image into high energy regions. The highest energy regions are assumed to contain target scattering centers of interest, while the lower energy regions are assumed to contain predominately background clutter scattering. Since the scattering model in Eqn. 4 does not effectively model clutter behavior with low model order, clutter energy must be suppressed in order to ensure low-variance parameter estimates.

Second, each segmented high-energy image region is assumed to be electrically isolated from other peak regions. We therefore process each region in parallel with a local model order that describes the number of scattering centers in the region only. A reduction in estimation complexity is therefore achieved by the divide-and-conquer approach to model order; model for an entire image (or image chip) can be quite large, while an individual peak region may have local model order of five or less.

Third, image domain processing of each peak region reduces computation complexity by reducing the number of pixels considered in the least-squares fit of the parametric model to the measured data. Fourth, image domain processing allows insertion of the model-based scattering analysis into a multi-staged automatic target recognition algorithm. The model-based scattering analysis is performed only after a computationally inexpensive prescreening stage [Novak, et al., 1993], and the image domain representation conveniently combines both motion compensation data and antenna measurements.

The model in Eqn. 4 describes scattering in the frequency-aspect domain. In order to accomplish image domain parameter estimation, we transform the scattering model from the frequency-aspect domain into the image domain. We process the parametric model using the same series of operations through which the motion-compensated frequency-aspect angle measurements would pass during image formation. There are many methods for image formation [Mensa, 1991], but we limit the discussion in this work to the two-dimensional Inverse Fourier Transform (IFT) of the measured frequency-aspect data. This imaging algorithm is widely used in SAR systems for which the center frequency of the radar is large compared to the bandwidth of the radar. In order to transform the frequency-aspect model into the image domain, we analytically perform a two-dimensional Inverse Fourier Transform on the scattering model of Eqn. 4.

We begin with the model in Eqn. 4.

$$\begin{aligned} E_n^s(f,\phi) &= A_n \left(j\frac{f}{f_c}\right)^{\alpha_n} \\ &\quad \cdot\mathrm{sinc}\left(\frac{2\pi f}{c}L_n \sin(\phi-\phi_n)\right) \\ &\quad \cdot\exp(-2\pi f \gamma_n \sin\phi) \\ &\quad \cdot\exp(-j\frac{4\pi f}{c}(x_n\cos\phi + y_n \sin\phi)) \end{aligned}$$

First, we replace the power dependence of amplitude on frequency with an exponential, as in [Chiang, 1996]:

$$\left(\frac{2\pi f}{c}\right)^{\alpha_n} \approx \exp\left(-2\pi r_n f\right) \qquad (6)$$

where r_n is a damping factor. We let the j^α term be absorbed into the complex amplitude, A_n. We adopt the following affine map from r_n to α_n, as proposed in [Chiang, 1996].

$$\alpha_n = \frac{f_c}{\Delta f}\left\{\exp\left(-2\pi\Delta f r_n\right) - 1\right\} \qquad (7)$$

The expression in Eqn. 7 is extremely accurate for small relative bandwidths [Chiang, 1996]. For example, at ten percent relative bandwidth the approximation in Eqn. 6 has less than 0.0001% relative error. As the bandwidth increases this error increases. Using this approximation, we first estimate r_n and then map it to α_n.

Second, we translate the model from polar coordinates to Cartesian coordinates via the substitution

$$\begin{aligned} f_x &= f\cos\phi \\ f_y &= f\sin\phi \end{aligned} \qquad (8)$$

By making this coordinate transformation to the Cartesian frequency plane, we assume that the measured data is sufficiently narrow in bandwidth so as to allow simple, approximate interpolation [Munson, et al., 1983] to a rectangular grid. We further approximate

$$2\pi f r_n \approx 2\pi f_x r_n \qquad (9)$$

in the frequency-dependent exponential of Eqn. 6; this approximation is valid for small angle spans.

Third, frequency and angle domain window functions are used in SAR imaging for sidelobe suppression. We assume that the window functions are separable in their Cartesian components and can be written as

$$\begin{aligned} W(f_x,f_y) &= W_x(f_x)\,W_y(f_y) \\ W_x(f_x) &= \sum_{p=1}^{P} B_p^x \exp\left(j2\pi\beta_p^x f_x\right) \\ W_y(f_y) &= \sum_{q=1}^{Q} B_q^y \exp\left(j2\pi\beta_q^y f_y\right) \end{aligned} \qquad (10)$$

We note that many commonly used window functions such as Rectangular, Hamming, and Taylor windows can be exactly written as in Eqn. 10. Inclusion of window parameters in the model adds the versatility needed for cases where only image domain data is available and the effect of the window is present in the image. The cost of including the window function is increased model complexity.

Fourth, we transform $E_n^s(f_x,f_y)$ to the image domain with a two-dimensional inverse Fourier Transform. Note that in practice measured data exists at a finite number of discrete frequencies and aspect angles. As a result, the IFT performed to generate radar imagery is typically an Inverse Discrete Fourier Transform (IDFT). Here we analytically perform a continuous IFT for simplicity. In fact, the alternative image domain model using the IDFT is not available in closed form. The IDFT is approximately equal to the continuous IFT when the image domain signal is essentially support-limited. Since most radar imagery contains a small number of high energy regions that are limited in extent, the sampling-induced aliasing is negligible. Thus, we assume that the sampled IDFT is well-approximated by a continuous IFT for radar imagery.

The image domain model $e_n^s(t_x,t_y)$ for a single scattering center is then written as

$$e_n^s(t_x,t_y) = \int_{f_{y1}}^{f_{y2}}\int_{f_{x1}}^{f_{x2}} \left\{\sum_{p=1}^{P} A_n B_p^x B_q^y \right.$$

$$\cdot\mathrm{sinc}\left[\frac{2\pi L\cos\phi_t}{c}\left(f_y - f_x\tan\phi_t\right)\right]$$

$$
\begin{aligned}
&* \exp\left[-2\pi f_y\left(\gamma_n + j\left(\frac{2y_n}{c} - \beta_q^y - t_y\right)\right)\right] \\
&* \exp\left[-2\pi f_x\left(r_n + j\left(\frac{2x_n}{c} - \beta_p^x - t_x\right)\right)\right] \\
&\qquad\qquad\qquad\qquad \left. df_x df_y \right\} \quad (11)
\end{aligned}
$$

where $*$ denotes convolution, f_{x1}, f_{x2} are the first and last f_x frequencies, and f_{y1}, f_{y2} are the first and last f_y frequencies.

As discussed in Section 2, either $L_n = 0$ or $\gamma_n = 0$, and we consider each case separately. If $L_n = 0$, evaluation of the integrals in Eqn. 11 yields

$$
\begin{aligned}
e_n^s(t_x, t_y) &= A_n \sum_{p=1}^{P}\sum_{q=1}^{Q} B_p^x B_q^y F_x F_y \\
&\cdot \exp\left(-2\pi f_{xc}\left(r_n + j\left(\frac{2x_n}{c} - \beta_p^x - t_x\right)\right)\right) \\
&\cdot \exp\left(-2\pi f_{yc}\left(\gamma_n + j\left(\frac{2y_n}{c} - \beta_q^y - t_y\right)\right)\right) \\
&\cdot \text{sinhc}\left(\pi F_x\left(r_n + j\left(\frac{2x_n}{c} - \beta_p^x - t_x\right)\right)\right) \\
&\cdot \text{sinhc}\left(\pi F_y\left(\gamma_n + j\left(\frac{2y_n}{c} - \beta_q^y - t_y\right)\right)\right) \qquad (12)
\end{aligned}
$$

where

$$
\begin{aligned}
F_x &= f_{x2} - f_{x1} \\
F_y &= f_{y2} - f_{y1} \\
f_{xc} &= \frac{f_{x2} + f_{x1}}{2} \\
f_{yc} &= \frac{f_{y2} + f_{y1}}{2} \\
\text{sinhc}(x) &= \frac{\sinh(x)}{x}
\end{aligned}
$$

The model in Eqn. 12 is for $L = 0$, which corresponds to a localized scattering mechanism. In the image domain, the localized mechanism is represented by two separable functions in t_x and t_y, each of which appears as a sinhc(x) function.

On the other hand, if $\gamma_n = 0$, then evaluation of the integrals in Eqn. 11 yields

$$
\begin{aligned}
e_n^s(t_x, t_y) &= \sum_{p=1}^{P}\sum_{q=1}^{Q} B_p^x B_q^y \frac{c}{j8\pi^2 L \cos\phi_t} \qquad (13) \\
&\cdot \exp(-2\pi f_{xc}\nu) \\
&* \{\exp(\pi F_x \nu)[I_2(K_1) - I_2 2(K_2)] \\
&+ \exp(-\pi F_x \nu)[I_2(K_3) - I_2(K_4)]\}
\end{aligned}
$$

where $I_2(K)$ is defined in the Appendix and

$$
\begin{aligned}
K_1 &= 2\pi(f_{y2} - f_{x1}\tan\phi_t) \\
K_2 &= 2\pi(f_{y1} - f_{x1}\tan\phi_t) \\
K_3 &= 2\pi(f_{y1} - f_{x2}\tan\phi_t) \\
K_4 &= 2\pi(f_{y2} - f_{x2}\tan\phi_t) \\
\nu &= r_n + j\left\{\frac{2x_n}{c} - \beta_p^x - t_x\right. \\
&\qquad \left. - \tan\phi_t\left(t_y - \frac{2y_n}{c} + \beta_q^y\right)\right\}
\end{aligned}
$$

As noted above, several approximations are made in arriving to Eqns. 12 and 13. We verify these approximations by comparing the image domain model to an image directly formed by applying the IDFT to a 128 × 128 array of polar-format samples given in Eqn. 4. The parameters chosen for this example verification are $L = 10m$, $\alpha = 1$, $x = 11m$, $y = 12m$, $\phi = 1.0°$ and $A = 1$. The relative error between the image domain model and the transformed frequency-aspect domain data for this example is less than 0.1%. This implies that the several approximations made in obtaining the parametric model in the image domain do not contribute significant error.

4 Curve Fitting

In this section we present an approximate Maximum Likelihood (ML) technique for estimating the parameters of the image domain scattering model. For each of p scattering centers, there are eight real-valued parameters to be estimated: the amplitude and phase, A_n, frequency damping r_n, aspect damping γ_n, length L_n, tilt angle ϕ_n, down range position x_n, and cross range position y_n. For the case where $L \neq 0$, we require $\gamma_n = 0$, whereas $L = 0$ implies ϕ_n is not estimated.

Application of the image domain model requires several radar sensor and image processing parameters. These image formation parameters are center frequency, bandwidth, total angle span (aperture), numbers of (interpolated) f_x and f_y frequency and angle samples, size of any zero-padding used in the IFFT, and the window functions used in down range and cross range. We assume that the image is generated using the 2-D IFFT of the measured frequency-aspect data. Further, we assume that the bandwidth of the radar data is sufficiently narrow to allow for the approximation of the polar frequency-aspect data as lying on a rectangular f_x, f_y grid. If this is not the case, it is assumed that a 2-D interpolation to the Cartesian frequency plane is done prior to the IFFT.

The initial step in our algorithm is to segment the image into small image chips, each of which contains a small number of scattering centers. Using the image domain model, a curve fit is then computed for each image chip. There exist automatic segmentation algorithms [Stach and LeBaron, 1996]; alternatively, the image can be segmented visually with human interaction. Whichever segmentation procedure is chosen, the result is a partitioning of the image into a set of smaller image chips, each of which contains very few scattering centers. The segmentation highlights an advantage estimating parameters in the image domain: we partition the large problem of estimating a single parametric model of large order to explain the entire data set into smaller, more tractable problems that can be solved in parallel.

For each segmented image chip we estimate the parameters of the scattering by computing a least-squares fit of the image domain model. For independent, identically distributed, zero mean additive Gaussian noise, the ML estimate of the parameters is found by minimizing the squared error between the model and the measured image domain data

$$J(\Theta) = \sum_{pixels} |\text{image chip} - \text{model}(\Theta)|^2 \quad (14)$$

where Θ is a vector containing the parameters to be estimated. An iterative optimization procedure is used to minimize Eqn. 14. There are many nonconvex optimization procedures in the literature, and we choose to use the simplex downhill method. The simplex method is desirable because it is numerically stable and does not require a gradient or Hessian of the cost function.

The least-squares cost function in Eqn. 14 is nonconvex with many local minima. Therefore, parameter initialization and model order selection [Sabharwal, et al., 1996] are very important. Presently, model order selection and the detection of $L \neq 0$ is performed interactively by a human user. Likewise, L_n and ϕ_n are initialized by the user. Initialization of range and cross range positions is computed from local maxima in the image chip, while r_n and γ_n are initialized at zero (point scattering). For a fixed parameter set Θ,the least-squares cost function J is quadratic in the complex amplitude parameter A; therefore, the least-squares estimate of A is computed noniteratively using a matrix pseudo-inverse.

At convergence, the simplex downhill optimization yields estimates of scattering parameters that describe the position, size, shape and orientation of the scattering centers that comprise the measured target. Automation of model order selection and parameter initialization is a topic of continuing development, both for our proposed scattering model and for simpler point scattering models [DeGraaf, 1997].

5 Statistical Analysis

In this section we investigate the statistical properties of the parametric scattering model presented in Eqn. 4. We use the Cramér-Rao bound (CRB) to provide a lower bound on the error variance of the estimated model parameters. These bounds are algorithm independent and provide an analysis of the uncertainty in the estimated parameters that describe the measured scattered field.

The utility of the variance bounds is twofold. First, we use the bounds to predict performance of an unbiased, statistically efficient estimator. Accordingly, the uncertainty in the estimated parameters can be characterized as a function of system parameters such as bandwidth, center frequency, SNR, frequency sample spacing, and scattering object. Second. we use the bounds to evaluate suboptimal, but computationally attractive, estimation algorithms. Specifically, the bounds provide a baseline for evaluating the trade-off between estimation performance and computation.

To derive bounds on estimation accuracy, we adopt the scattering model of Eqn. 4 with an additive perturbation

$$E(k, \phi) = \sum_{n=1}^{p} E_n^s(f, \phi) + \eta(k, \phi) \qquad (15)$$

Here, $\eta(k, \phi)$ represents the modeling error (background clutter, sensor noise, model mismatch, incomplete motion compensation, antenna calibration errors, etc.) and is assumed to be a white, Gaussian noise process. Recall L_n and ϕ_n are the length and orientation angle of a distributed scatterer, respectively, while x_n and y_n represent the down range and cross range position of the nth scattering mechanism. For this analysis, the model parameters are considered deterministic. Further, the performance predictions assume a parameter estimator that is unbiased, efficient and normally distributed (as is asymptotically true for the least-squares estimator).

The Cramér-Rao bounds are derived by a straight-forward application of the Cauchy-Schwartz inequality. We formulate the likelihood function of the scattering model under the assumption of additive, independent, identically distributed, complex-valued Gaussian noise; we then derive expressions for the partial derivatives of the logarithm of the likelihood function with respect to the parameters of the model and the variance of the noise. We use these derivatives and the independence of noise samples to obtain the Fisher information matrix.

The Fisher information matrix can be computed for any choice of scattering parameters and noise variance; the CRB covariance matrix is then found by inverting the Fisher information matrix. Finally, for any parameter in the set of scattering model parameters, a diagonal entry of the CRB covariance matrix gives a lower bound on the variance achievable by any unbiased estimator. Off-diagonal entries of the CRB matrix lower-bound the correlation between estimated parameters.

In this presentation, we use the CRB sensitivity analysis to address three performance issues relating SNR, bandwidth, and center frequency to parameter uncertainty: (1) resolution and accuracy in locating scattering mechanisms; (2) uncertainty in characterizing the frequency dependence of scattering; (3) accuracy in estimating the length and tilt of a distributed scatterer. These three issues are representative of the estimation performance predictions accessible via the statistical analysis.

For each of the examples presented below, we consider a bandwidth and aperture consistent with 1 ft × 1 ft SAR image resolution. Specifically, we assume 500 MHz bandwidth with ±1.4° aperture at $f_c = 10\,\text{GHz}$ and ±0.4242° aperture at $f_c = 33\,\text{GHz}$. Additionally, examples are computed for 64 equally spaced samples in both frequency and aspect. Signal-to-noise (SNR) values are reported as the ratio of signal to noise energy computed in the frequency-aspect domain samples; interpretation of SNR in the image domain as a difference between peak signal and clutter floor (*i.e.*, after pulse compression) requires a shift of 36 dB.

In Figure 1 we show the 95% confidence regions for the location estimates of two point scatterers separated by $10\sqrt{2} = 14.14\,\text{cm}$ ($f_c = 10\,\text{GHz}$). Adopting Hamming weighting for side lobe suppression, the standard image resolution cell is 18.10 cm; however, resolution of a model-based scattering analysis is limited only by sensor bandwidth, signal-to-clutter ratio, and model fidelity. In the figure, the 95% confidence regions are shown for -20 dB and 4 dB SNR; the circular regions reflect the uncorrelated errors in range and cross range estimates. A convenient definition of resolution is to require nonoverlapping confidence regions; here, coherent processing of the two-dimensional (k, ϕ) data resolves the two point scatterers for any SNR exceeding $-19\,\text{dB}$. In contrast, one-dimensional processing of a single frequency scan requires +19 dB SNR for 14.14 cm resolution (*i.e.*, 4σ separation).

In Figure 2 we show the probability of correct detection of the frequency dependence parameter, α, for a single scattering mechanism. Figure 2(a) shows detection rate versus SNR for 1 ft resolution X-band and K-band SAR systems ($f_c = 10\,\text{GHz}$ and $f_c = 33\,\text{GHz}$). The analytically derived detection results are averaged over five scattering types ($\alpha \in \{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$).

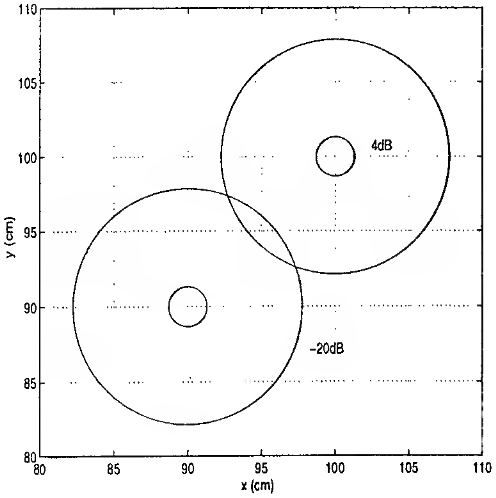
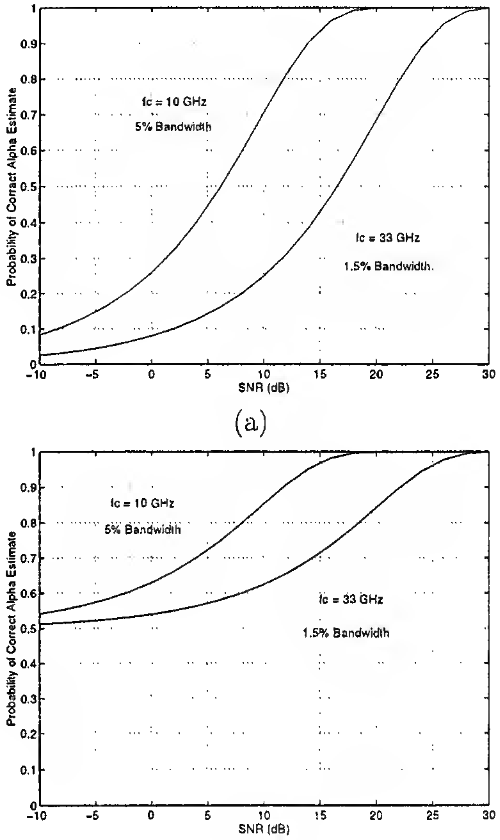


Figure 1: Resolution, as defined by 95 percent confidence regions on estimated scattering center locations. Fourier resolution is 18.1 cm.

Notably, uncertainty in estimating α decreases drastically with an increase in relative bandwidth. This finding confirms the intuition that accurate estimation of the trend in scattering amplitude versus frequency requires either high bandwidth or low noise power. In Figure 2(b) the detection of α is restricted to the binary hypotheses of $\alpha = \frac{1}{2}$ or $\alpha = 1$; this represents, for $L \neq 0$, the scenario of distinguishing a cylinder from a dihedral.

In Figure 3 we show the standard deviation of parameter estimates versus SNR for a dihedral near broadside ($L = 5\,m$, $\alpha = 1$, $\phi_t = 0.5°$, $f_c = 10\,\text{GHz}$, 500 MHz bandwidth). Observe that the down range and cross range uncertainties in the location of the dihedral are not equivalent. Appealing to intuition in the image domain, the nonsymmetric resolution is explained in that a sharp peak down range is more easily located than is the center of the broad cross range response. As the tilt angle, ϕ_t, increases, the difference between the cross range and down range location uncertainties decreases. This affect is easily understood by considering that the dihedral response, measured off of normal incidence, is dominated by diffraction from the two endpoints of the dihedral. Thus, off of normal incidence, the dihedral response is well approximated by two individual scattering mech-



(a)

(b)

Figure 2: Predicted probability of correctly identifying alpha: (a) five alpha values, (b) alpha $\frac{1}{2}$ versus 1

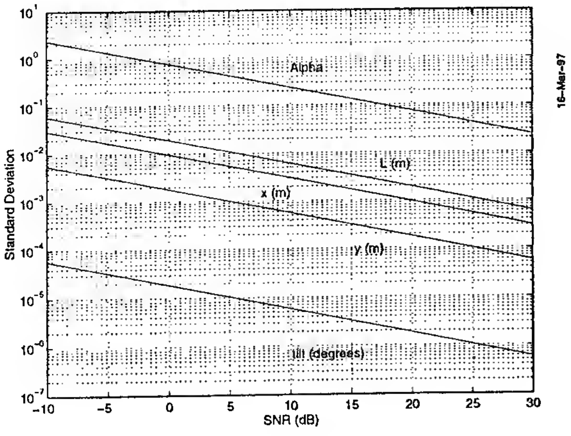


Figure 3: Parameter uncertainty versus SNR for scattering model, $L \neq 0$.

anisms, each with symmetric down range and cross range resolvability.

6 Examples

We present two measured target examples which illustrate the effectiveness of our image domain model at compressing large measured data sets into a small set of physically descriptive parameters. These parameters describe the shape, position, and orientation of scattering centers comprising the target response over the measured frequency and angle spans.

First we consider the scattering from a square flat plate measured in the Ohio State University ElectroScience Laboratory (ESL) Compact Range [Walton and Young, 1984]. We analyze stepped frequency measurements of the plate for frequencies 9.5–10.5 GHz in 20MHz steps and for angles ±3 degrees (in 0.5 degree steps) from broadside to one of the edges. The plate is a two foot square and lies in the plane of rotation. The measurement polarization is horizontal.

Figure 4 shows an image of the plate. The image contains three scattering centers. The broadside response of the edge of the plate appears as a line in the image. The two remaining corners on the back of the plate appear as point mechanisms. These three mechanisms are segmented in the image, and the algorithm of Section 5 is used to estimate the parameters. Table 3 shows the estimated parameters and their actual values The actual values are based on the assumption that the plate is exactly two foot square and is perfectly aligned during radar measurements so that zero degrees corresponds to broadside to an edge. The estimated tilt angle is approximately −0.6 degrees, which is an indication that the plate was not exactly aligned with 0 degrees broadside to the radar. Figure 5 shows the amplitude of the scattering from the plate as a function of angle. Note that the peaks are not at zero degrees and 90 degrees, as we would expect for a perfectly aligned target. The misalignment of the target also contributes to a small amount of error in the expected locations of the three scattering centers.

Table 3: Estimated scattering parameters for plate example; Fourier resolution, with Hamming windowing, is 19.2 cm.

Scatterer	Attribute	Estimated	Actual
Front Edge	length	0.5920m	.6096m
	tilt	−0.6567°	0
	down range	-0.3085m	-0.3048m
	cross range	-0.0014m	0.0000m
	alpha	0	0
Back Left Corner	down range	0.3048m	0.3048m
	cross range	-0.3157m	-0.3048m
	alpha	-1	-1
Back Right Corner	down range	0.2971m	0.3048m
	cross range	0.3216m	0.3048m
	alpha	-1	-1

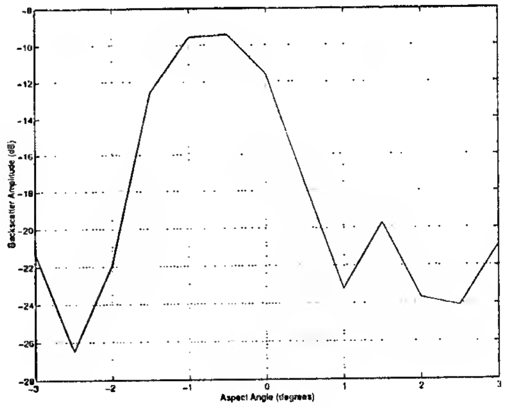


Figure 5: Magnitude of Plate scattering vs. aspect angle indicating target misalignment

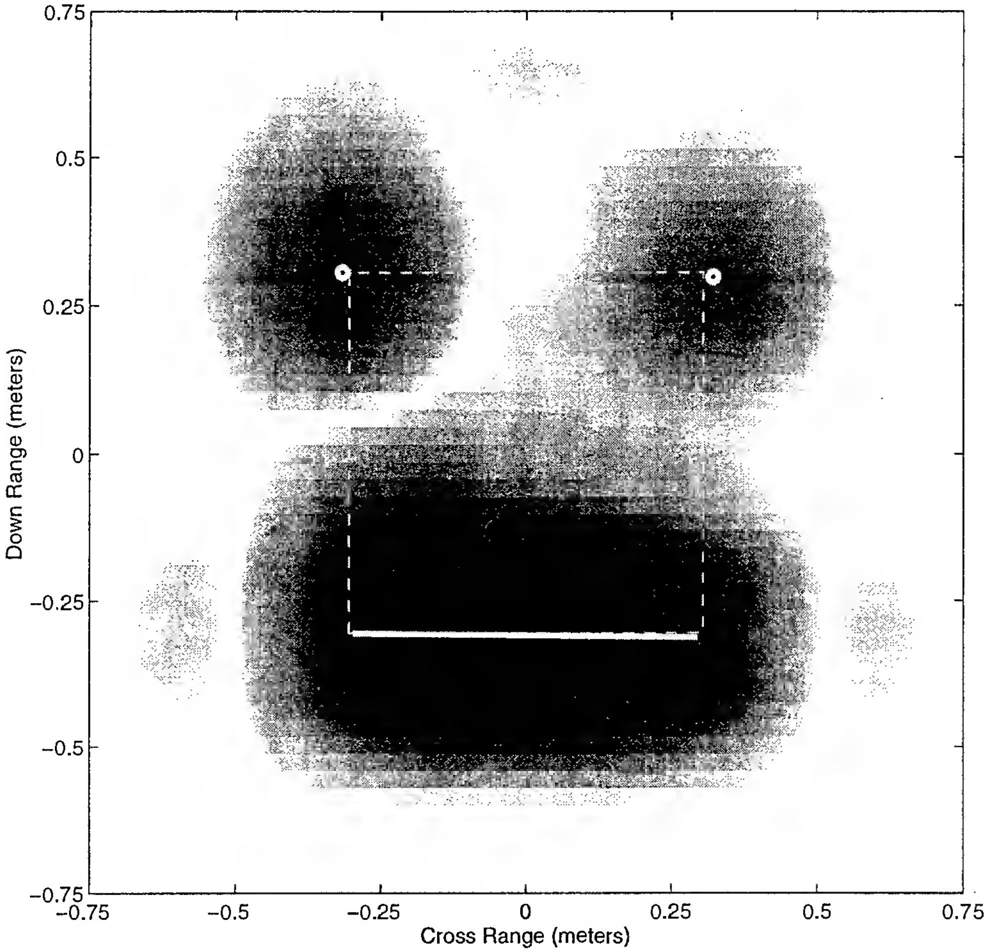


Figure 4: Image and estimates for plate example

The algorithm successfully compresses the measured data set into a small table of numbers describing three distinctive features of the plate. Over 97 percent of the measured energy is modeled by the three estimated scattering centers. Thus, the scattering model provides a 70:1 lossy compression of the original frequency-aspect data with a mean-squared image error (MSE) of three percent. The error in the estimated location of the individual scattering centers is small, and in each case is less than 1/10 the Fourier resolution. The geometric type (α) estimates correctly identify the edge specular and corner diffraction scattering behaviors.

Next we consider a more complicated target, namely a scale model of an M35 truck. Stepped frequency measurements of the 1:16 brass scale model truck were collected in the ESL compact range. As in the first example, we analyze data from 9.5–10.5 GHz and ±3 degrees (from normally incident on the back of the truck). Figure 6 shows the image of the truck for this data set. We segment the single most energetic scattering mechanism in the image and fit the line mechanism ($L \neq 0$) model to that image chip. Table 4 shows the estimated parameters for the model truck. The scattering center is estimated to have an alpha value of 1 which implies a specular surface, as is indeed present on the back of the model truck. The simulated image generated with the estimated parameters for the single mechanism represents the entire image with only six percent MSE at this frequency and angle span. Thus the model provides a 189:1 lossy compression of the original frequency-aspect data (4600:1 compression of the SAR image) with over 0.94 correlation to the original image.

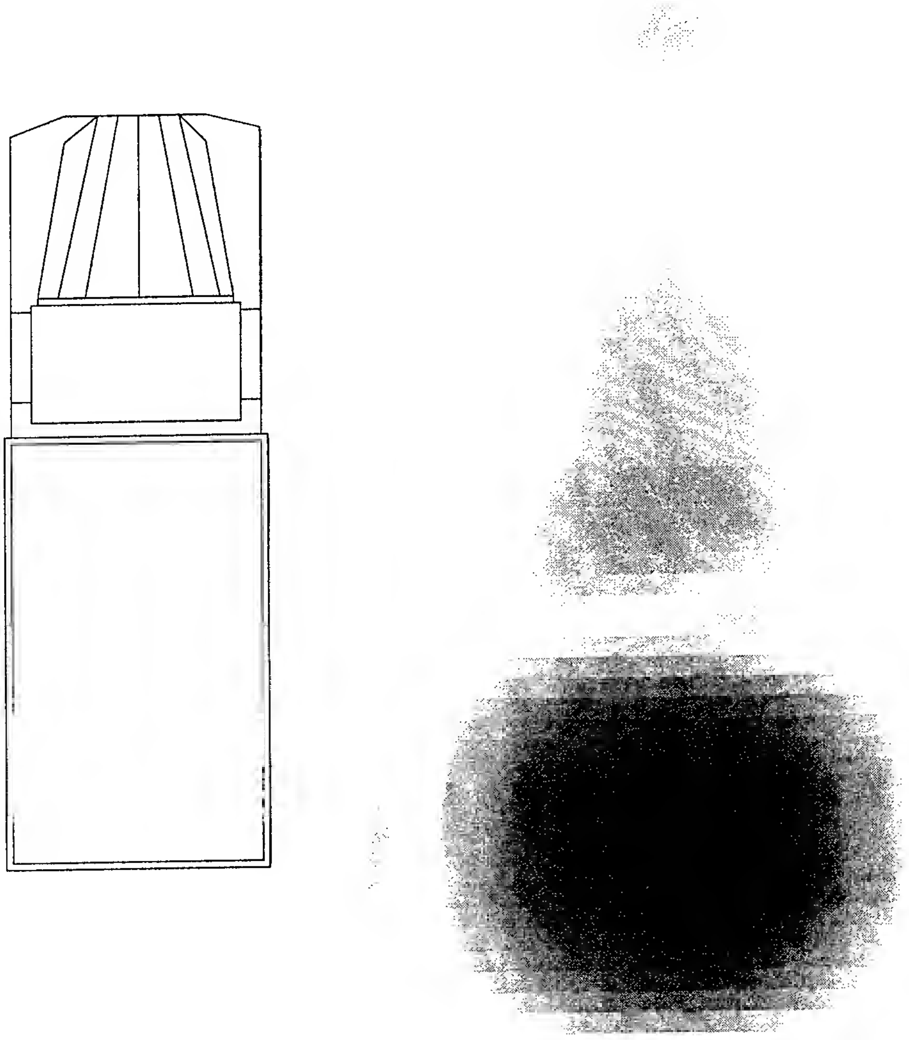


Figure 6: Image of scale model M35 truck

Table 4: Estimated scattering parameters for truck example

Scatterer	Attribute	Estimate
Back of Truck	Length	0.1595m
	Tilt	−0.4671°
	Down Range	-0.1502m
	Cross Range	-0.0033m
	Alpha	1

7 Conclusion

We present a GTD-based parametric scattering model for the extraction of scattering centers from radar data measured as a function of frequency and aspect angle. The scattering model balances physical fidelity with simplicity in functional form to yield both smaller modeling error and a richer description of scattering behavior when compared to either Fourier imaging or point scattering models. Data analysis using the proposed model has application to feature extraction for target identification, SAR data compression, and scattering studies. The model is developed in the frequency-aspect domain motivated by GTD-based and physical optics scattering principles. We present an image domain estimation procedure for the model parameters, and thereby gain benefit of both clutter suppression and computational savings. We derive Cramér-Rao bounds as tools for predicting uncertainty in estimated parameters. The scattering model and the image domain estimation algorithm are validated with two measured data examples.

Acknowledgments

The authors thank Dr. Jon Young of the Ohio State University ElectroScience Laboratory for use of the compact range anechoic chamber measurements of the 1:16 scale model M35.

References

[Chiang, 1996] Da-Ming Chiang, *Parametric Signal Processing Techniques for Model Mismatch and Mixed Parameter Estimation.* PhD thesis, The Ohio State University, Columbus, OH, 1996.

[DeGraaf, 1997] S. R. DeGraaf, "SAR Imaging via Modern 2-D Spectral Estimation Methods," *IEEE Trans. Image Processing*, to appear.

[Keller, 1962] J. B. Keller, "Geometrical Theory of Diffraction," *J. Opt. Soc. Am.*, vol. 52, pp. 116–130, 1962.

[Kouyoumjian and Pathak, 1974] R. G. Kouyoumjian and P. H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface," *Proc. IEEE*, vol. 62, pp. 1448–1461, November 1974.

[Mensa, 1991] D.L. Mensa, *High-Resolution Radar Cross Section Imaging.* Boston, MA: Artech House, 1991.

[Munson *et al.*, 1983] D. C. Munson, J. D. O'Brian, and W. K. Jenkins, " A Tomographic Formulation of Spotlight-Mode Synthetic Aperture Radar," *Proc. IEEE*, vol. 71, pp. 917–925, August 1983.

[Novak *et al.*, 1993] L. M. Novak, G. J. Owirka, and C. M. Netishen, "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System," *The Lincoln Laboratory Journal*, vol. 6, no. 1, pp. 11–24, 1993.

[Plonus *et al.*, 1978] M. A. Plonus, R. Williams and S. C. H. Wang, "Radar Cross Section of Curved Plates Using Geometrical and Physical Diffraction Techniques," *IEEE Trans. Antennas Propagat.*, vol. 26, pp. 488–493, May 1978.

[Potter *et al.*, 1995] L.C. Potter, D.-M. Chiang, R. Carriere and M.J. Gerry, "A GTD-Based Parametric Model for Radar Scattering," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 1058–1067, October 1995.

[Ross, 1966] R.A. Ross, "Radar Cross Section of Rectangular Flat Plates as a Function of Aspect Angle," *IEEE Trans. Antennas Propagat.*, vol. 14, pp. 329–335, May 1966.

[Sabharwal *et al.*, 1996] A. Sabharwal, C. J. Ying, L. Potter, and R. Moses, "Model order selection for summation models," *Proceedings of the Thirtieth Asilomar Conference on Signals, Systems, and Computers*, November 1996.

[Sacchini *et al.*, 1993] J. Sacchini, W. Steedly, and R. Moses, "Two-Dimensional Prony Modeling and Parameter Estimation ," *IEEE Trans. Sig. Proc.*, vol. 41, pp. 3127–3137, November 1993.

[Stach and LeBaron, 1996] J. Stach and E. LeBaron, "Enhanced Image Editing by Peak Region Segmentation," *Proceedings of 1996 AMTA Symposium*, October 1996.

[Tu *et al.*, 1997] M.-W. Tu, I.J. Gupta and E.K. Walton, "Application of Maximum Likelihood Estimation to Radar Imaging," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 20–27, January 1997.

[Walton and Young, 1984] E.K. Walton and J.D. Young, "The Ohio State University Compact Radar-Cross Section Measurement Range," *IEEE Trans. Antennas Propagat.*, vol. 32, pp. 1218–1223, November 1984.

Appendix

$$
\begin{aligned}
I_2(K) \;=\;& -j\cot\phi_t \exp\left(-K\cot\phi_t\left(r_n + j\left(\left(\frac{2x}{c} - \beta_p^x - t_x\right) + \tan\phi_t\left(\frac{2y}{c} - \beta_q^y - t_y\right)\right)\right)\right) \\
& \left(j2\pi\mathrm{sgn}\,(K)\,\mathrm{sgn}\,(-Kr_n\cot\phi_t)\,\mathrm{rect}_{\frac{L\cos\phi_t}{c}}\left(\cot\phi_t\left(t_x - \frac{2x}{c} + \beta_p^x\right)\right)\right. \\
+\;& E_1\left(-Kr_n\cot\phi_t - jK\left(\cot\phi_t\left(\frac{2x}{c} - \beta^x - t_x\right) - \frac{L\cos\phi_t}{c}\right)\right) \\
-\;& E_1\left(-Kr_n\cot\phi_t - jK\left(\cot\phi_t\left(\frac{2x}{c} - \beta^x - t_x\right) + \frac{L\cos\phi_t}{c}\right)\right)\Bigg) \\
+\;& j\pi\mathrm{sgn}\,(K)\,\mathrm{rect}_{\frac{L\cos\phi_t}{c}}\left(t_y - \frac{2y}{c} + \beta^y\right) \\
+\;& E_1\left(-jK\left(t_y - \frac{2y}{c} + \beta^y - \frac{L\cos\phi_t}{c}\right)\right) \\
-\;& E_1\left(-jK\left(t_y - \frac{2y}{c} + \beta^y + \frac{L\cos\phi_t}{c}\right)\right)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathrm{rect}_T\,(x) &= \begin{cases} 1, & -T/2 \le x \le T/2 \\ 0, & \text{otherwise} \end{cases} \\
E_1(z) &= \int_z^\infty \frac{\exp\,(-t)}{t}\,dt
\end{aligned}
$$

Multiple Stochastic Models for Recognition of Occluded Targets in SAR Images

Bir Bhanu and **Bing Tian**
College of Engineering
University of California, Riverside, California 92521
E-mail: bhanu@shivish.ucr.edu
URL: http://constitution.ucr.edu

Abstract

The correctness of results of structural object recognition approaches largely depends on the reliability of the features extracted from the image data. However, this cannot be satisfied in many practical situations where the applications require robust recognition during day/night under high clutter. Stochastic models provide some attractive features for pattern matching and recognition under partial occlusion and noise. In this paper, we present a hidden Markov modeling (HMM) based approach for recognizing objects in synthetic aperture radar (SAR) images. We develop multiple models for a given SAR image of an object and integrate these models synergistically using their probabilistic estimates for recognition. The models are based on sequentialization of scattering centers extracted from SAR images. Experimental results are presented using 99,000 training samples and 81,000 testing samples for 5 classes. We achieved better than 87% correct recognition performance when the objects are up to 30% occluded.

1 Introduction

One of the critical problems for object recognition is that the recognition process has to be able to handle partial occlusion of the object and spurious or noisy data. In most of the object recognition approaches, the spatial arrangement of structural information of the object is the central part that offers the most important information. Under partial occlusion situations the recognition process must be able to work with only portions of the *correct* spatial information. Rigid template matching and shape-based recognition approaches depend on good prior segmentation results. But the structural primitive (e.g., line segments, point-like features, etc.) extracted from occluded and noisy images may not have sufficient reliability, which will directly undermine the performance of those recognition approaches.

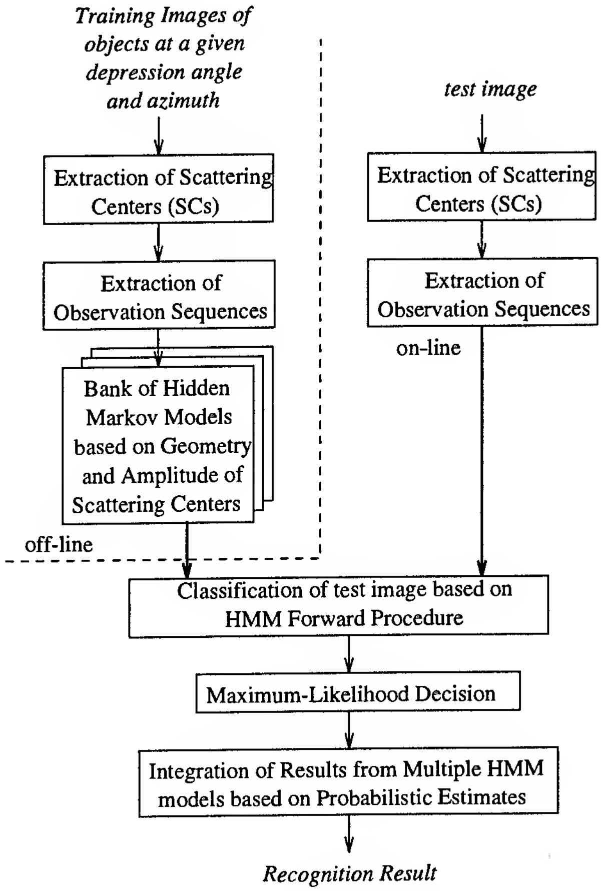
We want to suggest an object recognition mechanism that effectively makes use of all available structural information. Based on the nature of the problems caused by occlusion and noise, we view the spatial arrangement of structural information as a whole rather than view the spatial primitives individually. Because of its stochastic nature, a hidden Markov model (HMM) is quite suitable for characterizing patterns. Its nondeterministic model structure makes it capable of collecting useful information from distorted or partially unreliable patterns. Many successful applications of HMM in speech recognition [1, 2, 3] and character recognition [4, 5] attest to its usefulness. Thus, it is potentially an effective tool to recognize objects with partial occlusion and noise.

However, the limit of traditional HMMs is that they are basically one dimensional models. So how to appropriately apply this approach to two dimensional image problems becomes the key. It has been largely an unsolved problem. In this paper we use the features based on the image formation process to encode the 2-D image into 1-D sequences. We use information both from the relative positions of the scattering centers and their relative magnitude in SAR images [6]. In this paper we address the fundamental issues of building object models and using them for robust recognition of objects in SAR images.

1.1 Overview of the approach

Figure 1 provides an overview of the HMM based approach for recognition of occluded objects in SAR imagery. During an off-line phase, scattering centers are extracted from SAR images by finding local maxima of intensity. Both locations and magnitudes of these peak features are used in the approach. These features are viewed as *emitting patterns* of some hidden stochastic process. Multiple observation sequences based on both the relative geometry and relative amplitude of SAR return signal (obtained as a result of the physics of the SAR image forma-

*This work is supported by DARPA grant MDA972-93-1-0010. The contents and information do not necessarily reflect the position or the policy of the U.S. Government.



Training Images of objects at a given depression angle and azimuth
test image
Extraction of Scattering Centers (SCs)
Extraction of Scattering Centers (SCs)
Extraction of Observation Sequences
Extraction of Observation Sequences
on-line
Bank of Hidden Markov Models based on Geometry and Amplitude of Scattering Centers
off-line
Classification of test image based on HMM Forward Procedure
Maximum-Likelihood Decision
Integration of Results from Multiple HMM models based on Probabilistic Estimates
Recognition Result

Figure 1: The HMM-Based approach for recognition of occluded objects.

tion process) are used to build the bank of stochastic models to provide robust recognition in the presence of severe occlusion and unstable features caused by scintillation phenomena (where some of the features may appear/disappear at random in an image). At the end of the off-line phase, hidden Markov recognition models for various objects and azimuths are obtained. Similar to the off-line phase, during the on-line phase features are extracted from SAR images and observation sequences based on these features are matched by the HMM forward process with the stored models obtained previously. Maximum likelihood decision is made on the classification results. Now the results obtained from multiple models are combined in a voting kind of approach that uses both the object, azimuth label and its probability of classification. This produces a rank ordered list of classifications of the test image and associated confidences.

1.2 Related work and our contribution

There is no published work on object recognition using HMM models. Fielding and Ruck [7] have used HMM models for spatio-temporal pattern recognition to classify moving objects in image sequences. Rao and Mersereau [8] have attempted to merge HMM and deformable template approaches for image segmentation. Template matching [9] and major axis based approaches [10] have been used to recognize and index objects in SAR images, however, they are not suitable to recognize occluded objects. Recently, invariant histogram in conjunction with template matching have also been used to recognize occluded objects in SAR images [11].

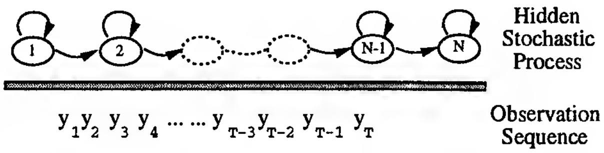
The original contributions of this paper are:

- Hidden Markov modeling approach commonly used for recognizing 1-D speech signals is applied in a novel manner to 2-D SAR images to solve the occluded object recognition problem.
- Multiple models derived from various observation sequences, based on both the geometry and signal amplitude are used to capture the unique characteristics of patterns to recognize objects.
- Unlike most of the work for model building in computer vision, our recognition models using hidden Markov modeling concept are based on the peculiar characteristics of SAR images where the number of models used for recognition is scientifically justified by the quantification of the azimuthal variance in SAR images.
- Extensive amounts of data (99,000 training samples and 81,000 testing samples obtained from 1800 images generated by the well known XPATCH SAR simulator [12] that uses 3-D CAD models of objects) is used to test the approach for recognition of objects for various amounts of occlusion (10−50%) and good recognition performance is obtained.

2 Hidden Markov Modeling Approach

It is well known that HMM can model speech signals well [1, 2, 3]. It is a model used to describe a doubly stochastic process which has a set of states, a set of output symbols and a set of transitions. Each transition is from state to state and associated with it are a probability and an output symbol. The word 'hidden' means that although we observe an output symbol, we cannot determine which transition has actually taken place. At each time step t, the state of the HMM will change according to a transition probability distribution which depends on the previous state and an observation y_t is produced according to a probability distribution which depends on the current state.

Formally, a HMM is defined as a triple $\lambda = (A, B, \pi)$, where a_{ij} is the probability that state i transits to state j, $b_{ij}(k)$ is the probability that we observe symbol k in a transition from state i to state j, and π_i


1
2
N-1
N
Hidden Stochastic Process
$y_1 y_2\ y_3\ y_4 \cdots\cdots y_{T-3} y_{T-2}\ y_{T-1}\ y_T$
Observation Sequence

N: the number of states.

M: the number of distinct observable symbols.

A: a_{ij} is the probability that state i will transit to state j.

B: $b_{ij}(k)$ is the probability that symbol k will be observed when there is a transition from state i to state j.

Π : π_i is the probability that state i is the initial state.

Figure 2: A N states forward-type HMM

is the probability of i being the initial state. Figure 2 shows an example of a N states HMM.

Recognition Problem – Forward Procedure: The HMM provides us a useful mechanism to solve the problems we face for robust object recognition. Given a model and a sequence of observations, the probability that the observed sequence was produced by the model can be computed by the forward procedure [13]. Suppose we have a HMM $\lambda = \{A, B, \pi\}$ and an observation sequence y_1^T. We define $\alpha_i(t)$ as the probability that the Markov process is in state i, having generated y_1^t.

$$\begin{aligned} \alpha_i(t) &= 0, \text{ when t=0 and i is not an initial state.} \\ \alpha_i(t) &= 1, \text{ when t=0 and i is an initial state.} \quad (1) \\ \alpha_i(t) &= \Sigma_j[\alpha_j(t-1)a_{ji}b_{ji}(y_t)], \text{ when } t > 0. \end{aligned}$$

The probability that the HMM stopped at the final state and generated y_1^T is $\alpha_{S_F}(T)$. After initialization of α, we compute it inductively. At each step the previously computed α is used, until the t reaches T. $\alpha_{S_F}(T)$ is the sum of probabilities of all paths of length T.

Usually, α will become too small to be represented in computer after several iterations. We take the logarithm of the α value in the computation.

Training Problem – Baum-Welch Algorithm: To build a HMM is actually an optimization of the model parameters so that it can describe the observation better. This is a problem of training. The Baum-Welch re-estimation algorithm is used to calculate the maximum likelihood model. But before we use the Baum-Welch algorithm, we must introduce the counterpart of $\alpha_i(t)$: $\beta_i(t)$, which is the probability that the Markov process is in state i and will generate y_{t+1}^T.

$$\begin{aligned} \beta_i(t) &= 0, \text{ when t=T and i is not a final state.} \\ \beta_i(t) &= 1, \text{ when t=T and i is a final state.} \quad (2) \\ \beta_i(t) &= \Sigma_j[a_{ij}b_{ij}(y_{t+1})\beta_j(t+1)], \text{ when } 0 \le t < T. \end{aligned}$$

The probability of being in state i at time t and state j at time $t+1$ given observation sequence y_1^T and the model λ is defined as follows:

$$\begin{aligned} \gamma_{ij}(t) &= P(X_t = i, X_{i+1} = j \mid y_1^T) \\ &= \frac{\alpha_i(t-1)a_{ij}b_{ij}(y_t)\beta_j(t)}{\alpha_{S_F}(T)} \qquad (3) \end{aligned}$$

Now the expected number of transitions from state i to state j given y_1^T at any time is simply $\Sigma_{t=1}^T \gamma_{ij}(t)$ and the expected number of transitions from state i to any state at any time is $\Sigma_{t=1}^T \Sigma_k \gamma_{ik}(t)$. Then, given some initial parameters, we could recompute $\overline{a_{ij}}$, the probability of taking the transition from state i to state j as:

$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^T \gamma_{ij}(t)}{\Sigma_{t=1}^T \Sigma_k \gamma_{ik}(t)} \qquad (4)$$

Similarly, $\overline{b_{ij}(k)}$ can be re-estimated as the ratio between the frequency that symbol k is emitted and the frequency that any symbol is emitted:

$$\overline{b_{ij}(k)} = \frac{\Sigma_{t:y_t=k} \gamma_{ij}(t)}{\Sigma_{t=1}^T \gamma_{ij}(t)} \qquad (5)$$

It can be proved that the above equations are guaranteed to increase $\alpha_{S_F}(T)$ until a critical point is reached, after which the re-estimate will remain the same. In practice, we set a threshold as the ending condition for re-estimation.

So the whole process of training a HMM is as follows:

1. Initially, we have only an observation sequence y_1^T and blindly set (A, B, π).
2. Use y_1^T and (A, B, π) to compute α and β (equations 1, 2).
3. Use α and β to compute γ (equation 3).
4. Use y_1^T, (A, B, π), α, β and γ to compute A and B (equations 4, 5). Go to *step* 2.

A HMM is able to handle pattern distortions and the uncertainty of the locally observed signals, because of its nondeterministic nature. However, a HMM is primarily suited for sequential, one-dimensional patterns and it is not obvious that how a HMM can be used on 2-D patterns in object recognition. The basic ideas to apply a HMM for our purpose are (a) training the HMM λ by samples of SAR images of a certain object, and (b) recognizing an unknown object in a given SAR image. These two problems are addressed in the following. The key questions are what we shall use as observation data and how we get the observation sequences.

3 Hidden Markov Models for SAR Object Recognition

3.1 Extraction of Scattering Centers

Scattering centers (location and magnitude) extracted from SAR images are used to train and test models for recognition. We consider a pixel as a scattering center if the magnitude of SAR return at this

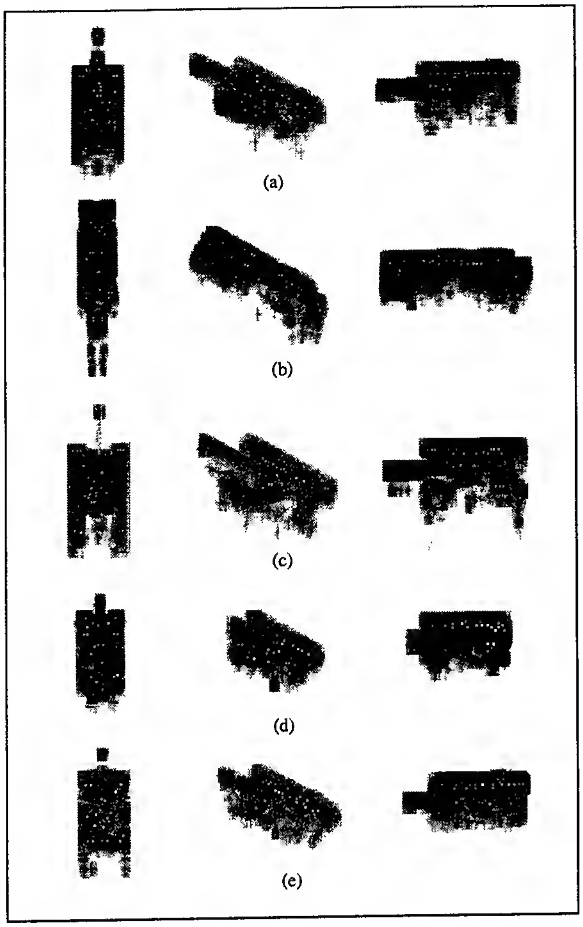
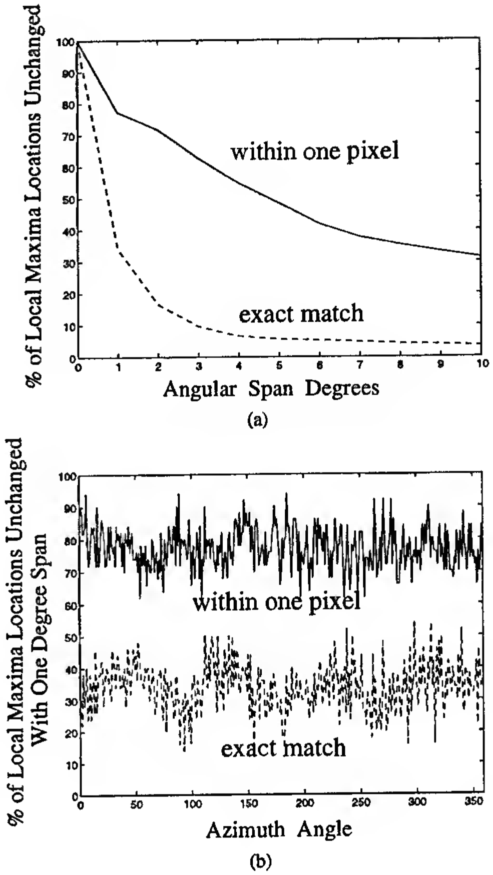


Figure 3: Examples of scattering centers (white dots) extracted from SAR images at azimuths $0°, 60°, 90°$. (a) Fred tank, (b) SCUD launcher with missile down, (c) T72 tank, (d) T80 tank, (e) M1a1 tank.

pixel is larger than all its eight neighbors. Figure 3 shows some examples of scattering centers extracted from SAR images (6″ resolution) of various objects at 15° depression angle and azimuths at 0°, 60°, and 90°.

3.2 Rotation Variance of Scattering Centers and Representation of 3-D Objects

Unlike the visible images, SAR images are extremely sensitive to slight changes in viewpoint (azimuth and depression angle) and are not affected by scale [14]. We evaluate the characteristics of scattering centers to find out what kind of location invariance exists among scattering centers. Figure 4(a) shows the rotation invariance for T72 tank. The data is obtained by rotating the image at azimuth $i°$ (for a fixed depression angle) by $x°$ (x from 1 to 10), and comparing the rotated image with the image of $(i + x)°$ to see how many scattering centers do not change their location. Since the object chip is 256 × 256 pixels, we rotate the image with respect to the center point (127.5, 127.5). The distance measurement criteria "exact match" and "within one pixel" are defined in the following:



(a)

(b)

Figure 4: (a)T72 tank Rotational Invariance.(b)T72 tank Rotational Invariance With 1° Angular Span.

$$\begin{cases} \mathbf{x_r} \textbf{ exactly matches } \mathbf{x}\textbf{:} \\ \text{if } MAX(|x - x_r|, |y - y_r|) < \frac{1}{2} \text{ pixel} \\ \mathbf{x_r} \textbf{ and } \mathbf{x} \textbf{ are within one pixel:} \\ \text{if } MAX(|x - x_r|, |y - y_r|) < 1\frac{1}{2} \text{ pixel} \end{cases}$$

Figure 4(a) shows the average result for images at all the 360 azimuth angles. The top 50 scattering centers are used for each image. Figure 4(b) gives the percentage of scattering center locations unchanged vs. azimuth angle with 1° angular span for the exact match and within one pixel match. These results show that scattering centers for SAR images vary greatly with relatively small changes of azimuth angles. As a result we represent an object at a given depression angle by 360 azimuths taken in steps of 1°.

3.3 Extraction of Observation Sequences

After the scattering centers are extracted, we need to encode the data into a 1-D sequence as the input to a recognition model based HMM process. It is one of the *key* factors which affects the performance

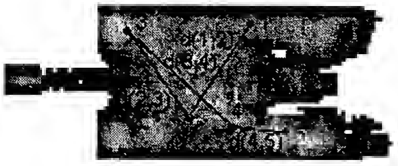


Figure 5: Example of an observation sequence superimposed on an image of T72 tank.

of a HMM modeling approach for object recognition. There are many ways to choose observation sequences, but we want to use information from both the magnitude and the relative spatial location of the scattering centers extracted from a SAR image. Also the sequentialization method should not be affected by distortion, noise, or partial occlusion and should be able represent the image efficiently.

Based on the above considerations, we employ two approaches to obtain the sequences.

- Sequences based on relative amplitudes: $O_1 = \{Magnitude_1, Magnitude_2, ..., Magnitude_n\}$
- Sequences based on geometrical relationship:
  $O_2 = \{d(1,2), d(2,3), ..., d(n,1)\}$ $(length\ n)$
  $O_3 = \{d(1,2), d(1,3), ..., d(1,n)\}$ $(length\ n-1)$
  $O_4 = \{d(2,1), d(2,3), ..., d(2,n)\}$ $(length\ n-1)$
  $O_5 = \{d(3,1), d(3,2), ..., d(3,n)\}$ $(length\ n-1)$

where $Magnitude_i$ is the amplitude of ith scattering center and $d(i,j)$ is the euclidean distance between scattering centers i and j. Figure 5 gives an example to illustrate how we get the sequences. Sequence O_1 is obtained by sorting the scattering centers by their magnitude. We label the scattering centers 1 through n in descending order. So in this approach, we do not use the location information and thus can avoid the instability caused by the error in localization of scattering centers. Sequences O_2 through O_5 are obtained based on the relative locations of the scattering centers. In experiments described in section 4, we only consider the top 20 scattering centers. This is because we expect that the scattering centers with larger magnitude are relatively more stable than the weaker ones.

Since we use discrete HMMs, each element in the sequence should be converted to an observation symbol. It is like a label from 1 to K that represents the symbols which can be observed for a HMM. We use the K-means algorithm [15] to classify the magnitude values (or distance values) of all the scattering centers in the database into K classes. Once we know to which class each of the elements of a sequence belongs, we label the element with the label of its class. Thus, the sequence of magnitude values (or distance values) now is changed to a label between 1 to K which represents how different scattering centers fall into the different groups and finally, for a given sequence, we obtain a sequence of observation symbols.

3.4 Off-line Training Phase

The procedure for building the model base is described as follows:

1. Loop (for a given depression angle) lines 2-4 for each object and each azimuth angle.
2. Generate images which simulate occlusion with scattering centers occluded from different directions (see section 4.1).
3. Loop line 4 for each image generated by line 2.
4. Use Baum-Welch algorithm to re-estimate the HMM parameters. (Exit $3-4$ loop when there is no further change in parameter values.)

3.5 On-line Recognition Phase

The recognition procedure is described as follows:

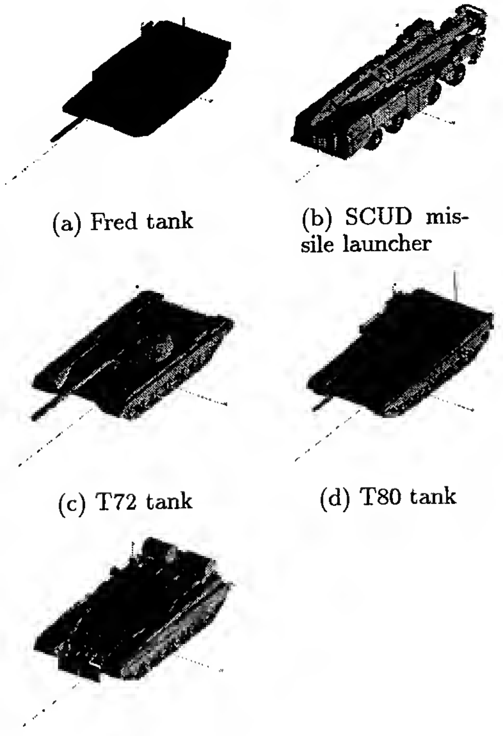
1. Loop lines 2-3 for all the testing observation sequences.
2. Loop line 3 for all the models in the model base.
3. Feed the observation sequence into the model, $(A, B, \Pi)_{(M_i^*, a_j^*)}$, Use Forward algorithm to compute the probability that this sequence is produced by this model.
4. The model with maximum probability of an observation sequence is selected as the best match.

4 Experiments

4.1 Data

Using the well known XPATCH [12] SAR simulator, we generate one set of SAR images of 5 objects (Fred tank, SCUD missile launcher, T72 tank, T80 tank and M1a1 tank, shown in Figure 6.) at 15° depression angle, at each of the azimuth angles from 0° to 359°. We extract the 20 scattering centers (local maxima) with largest magnitudes. In the experiments, since we want to test the performance of our approach under partial occlusion and spurious data, we simulate realistic occlusion situations and generate images for training and testing.

Simulating occlusion: We consider the occlusion to occur possibly from 9 different directions as shown in Figure 7. Scattering centers being occluded are not available, moreover, we add some spurious data into the image. For instance, 20 scattering centers are shown in each image of Figure 7. They are obtained by removing 4 scattering centers from one particular direction (simulated occlusion) and adding 4 spurious scattering centers into the image. The spurious scattering centers are added based on the following rules:



(a) Fred tank

(b) SCUD missile launcher

(c) T72 tank

(d) T80 tank

(e) M1a1 tank

Figure 6: Targets.

- The location of the scattering center is generated as a pair of random numbers.
- The magnitude of the scattering center depends on a random number r between 1 and 50. If r is between 1 and 20, we use the magnitude of the rth brightest image scattering center as the magnitude of the spurious center. Otherwise, we choose the magnitude of the 21st brightest scattering center if it was not already assigned to another spurious center. If it was already chosen, we will select the magnitude of the first unused scattering center (the 22nd, the 23rd, and so on).

Training Data: Based on the method of simulating occlusion described above, we generate 90 images from the original image (10 samples for each of 9 directions) at 5% occlusion and another 90 images at 10% occlusion. Including the original image, we have 181 images per object per azimuth angle to train multiple HMM models. Then we have a total of 99,000 (5 objects, 360 azimuths, 55 occluded images) samples for training.

Testing Data: We generate one image with o scattering centers occluded ($o = 2, 4, 6, 8$ or 10) from direction d ($d = 0, 1, ..., 8$) per azimuth angle per object. So there are 1800 images (5 objects × 360 degrees) generated for testing of occlusion with o scattering centers occluded from direction d. Thus, we have a total of 81,000 (5 objects, 360 azimuths, 5 different occlusions 10% – 50%, and 9 directions) samples for testing.

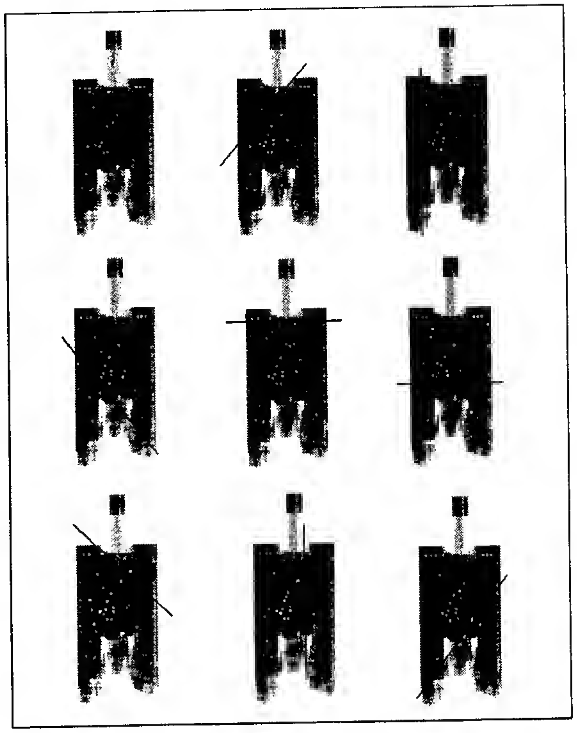


Figure 7: Scattering centers of T72 tank at azimuth 0°, part of scattering centers are occluded from a particular direction (0-8, left to right, top to bottom).

4.2 Training – Building Bank of HMM Models for Recognition

We performed experiments to choose the optimum of number of states and number of symbols of the HMM. We use data from 5 azimuth angles of five objects (Fred tank, SCUD missile launcher, T80 tank, T72 tank, and M1a1 tank). The results are shown in Table 1.

We find that with the increase in the number of states and symbols, recognition performance increases. Considering both the recognition performance and the computation cost, we choose 8 states and 32 symbols as the optimal number of states and symbols for our HMM models. Figure 8 illustrates example parameters of a 5 state, 4 symbol HMM.

We have 1800 (= 360 × 5) HMM models. Further, since we have defined five kinds of observation sequences for each image (O_1, O_2, O_3, O_4, O_5), we get models based on each kind of observation sequence.

Table 1: Recognition rate of HMM with different number of states and symbols.
N - # of states.
M - # of symbols.
R - Recognition rate % (top answer is correct).
I - Indexing rate % (correct answer is in the top 5).

		id only		*id with pose*	
N	M	R	I	R	I
4	8	76.1	96.5	62.6	79.9
4	16	89.6	98.4	85.4	93.1
4	24	95.1	99.3	91.8	97.3
4	32	96.6	99.9	94.8	99.0
4	64	99.7	100.0	99.6	100.0
5	8	80.1	97.4	67.3	84.0
5	16	91.9	98.6	86.7	93.7
5	24	96.6	99.7	94.6	98.6
5	32	97.8	99.8	96.7	99.3
5	64	99.9	100.0	99.9	100.0
6	8	82.5	96.9	71.7	84.8
6	16	93.8	99.5	90.1	96.7
6	24	98.5	99.8	97	99.7
6	32	98.9	100.0	98.5	99.9
6	64	100.0	100.0	100.0	100.0
8	8	84.3	97.6	77.4	87.6
8	16	96.4	99.8	94.6	98.3
8	24	99.4	100.0	99	99.9
8	32	99.8	100.0	99.8	100.0
8	64	100.0	100.0	100.0	100.0
10	8	100.0	100.0	100.0	100.0
10	16	98.3	99.9	97.3	99.6
10	24	99.9	100.0	99.9	99.9
10	32	100.0	100.0	99.9	100.0
10	64	100.0	100.0	100.0	100.0

4.3 Testing Results

During testing phase, each of the 81,000 testing images is tested against all models (1800 models: 5 objects, each has 360 models for each azimuth angle). If the model with the maximum probability is the model which produced the sequence, we count it as one correct recognition. Otherwise, we count it as one incorrect recognition. After we get the results of scattering centers occluded from all 9 directions, we average the result and associate this recognition performance with the model.

Figure 9 shows the testing results for each of the five kinds of sequences: $O_1, O_2, ..., O_5$ (section 3.3). The top curve, a dotted line, is the percentage that the test case object and pose is among the top ten recognition results, and the lower curve, in solid line, indicates the percentage that the recognition result with the highest probability is the same as the test case object and pose.

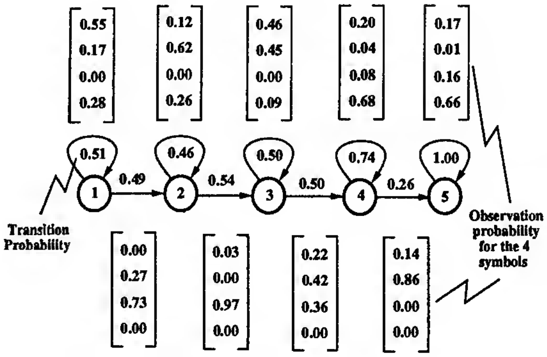


Figure 8: An example: parameters of a 5 states, 4 symbols HMM. The number on edges represents the transition probability, and the vector associated with each transition represents $b_{ij}(k)$. In our case, we use HMM with 8 states, 32 symbols

Integration of results from multiple sequences: Since not all models based on various sequences for a particular object and azimuth will provide optimal recognition performance under occlusion, noise, etc., we improve the recognition performance by combining the results obtained from all five kinds of models. Before discussing the approach for integration, we ask the question that if one testing image cannot be recognized correctly by models based on a particular sequence, say O_1, will it be recognized correctly by models based on other kinds of sequences?

From the testing results, we obtained Table 2 which shows how many incorrect recognitions, made by using models based on sequence O_2, can be correctly recognized ("captured") by models based on other sequences. We draw two curves (Figure 10(a)) to show the possible "upper bound" and "lower bound" of recognition rate we can achieve based on the 5 kinds of models. We *define* the "upper bound" as the highest possible recognition performance that can be achieved using the 5 models (O_1 to O_5) considering *only* the top candidate for recognition from each of the models. The curve on the top is obtained by considering all 5 kinds of models, if one of them can correctly recognize the test data, we count it as a correct recognition. The total number of errors corresponding to "upper bound" are shown in the 7th column of the Table. The "lower bound" or the bottom curve is the worst recognition result out of the five models.

We have developed a histogram-like method shown in Figure 11 to integrate the results from models based on 5 different sequences.

1. For each test image, we collect the ten highest possibilities in the testing results corresponding to each of the sequences $O_1, O_2, ..., O_5$.
2. A normalization is done to the ten probabilistic estimates corresponding to each of the sequences. So we have 50 normalized numbers for each test image.

Table 2: Testing results for occluded object recognition using of 81,000 testing cases.

Percent. occlusion	*Errors with model O_2*	*Errors Captured by models*				*Errors using models O_1 to O_5*	*% Correct Recognition ("upper bound")*	*% Based on Integration Recognition*	*% Based on Integration Indexing*
		O_1	O_3	O_4	O_5				
10%	4	0	1	0	1	2	100.0	99.9	99.9
20%	271	19	53	74	101	121	99.6	98.9	99.6
30%	763	111	294	339	418	144	98.6	93.4	97.6
40%	1050	265	580	629	675	79	95.6	79.4	91.8
50%	1119	397	726	755	784	37	91.8	62.2	83.3
Average Recognition Rate							97.1	86.8	94.4

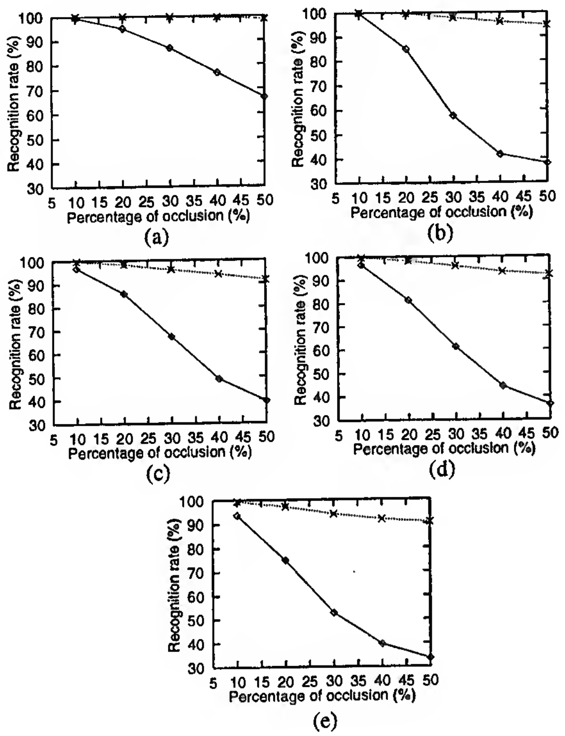


Figure 9: Recognition rate vs. percentage of occlusion for HMM models based on (a) O_1, (b) O_2, (c) O_3, (d) O_4, and (e) O_5.

3. We draw a histogram with probability vs. object and pose (here we combine object and pose as one parameter). This is because because each number corresponds to an object and a pose (the number is the probability that the test image is the image of that object at that pose),

4. If the object associated with the highest probability in the histogram is the same as the ground truth, we count it as one correct recognition.

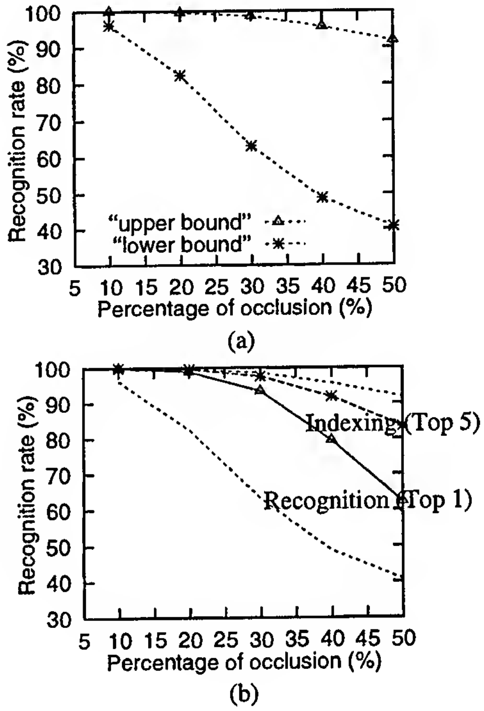


Figure 10: (a) "Upper" and "lower" bound of recognition rate vs. percentage of occlusion. (b)Performance of integrated models: using integrated models O_1 to O_5. The results for recognition (Top 1) and indexing (Top 5) candidates are superimposed on the figure shown in (a).

The second curve from the bottom in Figure 10(b) is the result. The corresponding confusion matrix for various amounts of occlusion is shown in Table 3. On the average, we find 80.35% correct recognition performance when the objects are occluded from 10 – 50%. The second curve from the top in Figure 10(b) is obtained by counting a correct indexing result when the ground truth is in the objects associated with the highest 5 probabilities in the histogram. For the purpose of comparison, we have also superimposed the curves in Figure 10(a) into Figure 10(b) with "lower/upper" bounds. Considering the correct indexing answer in the top 5 responses, the average performance is 93.3% for 5 objects occluded from 10% – 50%. Thus, our method of integration produces good results in comparison to "upper bound" which is 95.3% for 5 objects for 10% – 50% occlusion.

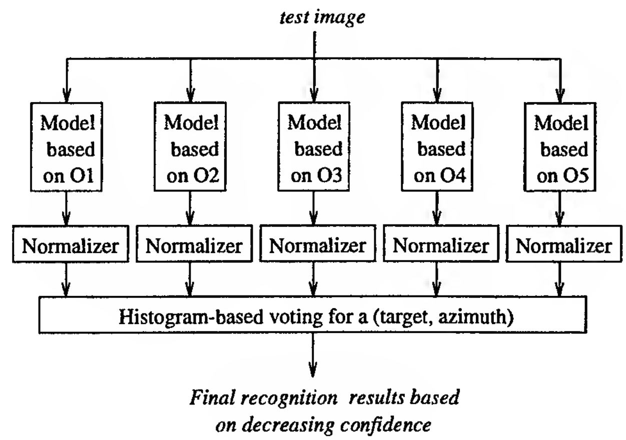


Figure 11: Integration of results by histogram-based method.

5 Conclusions and Future Work

We have presented a novel conceptual approach for the recognition of occluded objects in SAR images. The approach uses multiple HMM based models for various observation sequences that are chosen based on the SAR image formation and account for both the geometry and magnitude of SAR image features. Using 99,000 training samples and 81,000 testing samples, we find 86.76% average correct recognition performance on 5 classes of objects with 10% – 50% occlusion. The number of observation sequences and the number of features are design parameters which can be optimized by following the approach presented in the paper.

We have also done some initial experiments for articulated object recognition using HMM approach. We have three sets of data: the original images for the objects (T72 tank, T80 tank, and M1a1 tank), the images for the objects with turret at 60 degree articulation, and the images for the objects with turret at 90 degree articulation. We compared the observation sequences O_1 extracted from the three sets of images. Figure 12 shows the analysis graph for T72 tank. Figure 12 (a1), (b1), and (c1) are obtained by counting the number of observation symbols in observation sequence of one image which are the same as its corresponding one in observation sequence of another image. Figure 12 (a2), (b2), and (c2) are obtained by counting the sum of differences between observation symbols in observation sequence of one image and its corresponding one in observation sequence of another image.

We used two sets out of three sets of images as training data to train the HMM models, and tested the HMM models on the other set. Table 4 shows the results. These experimental results are obtained by using observation sequence O_1 only, the experiments using other sequences O_2 through O_5 will be done in the future.

Table 3: Confusion Matrix for 5 objects classes at varying amounts of occlusion (10% – 50%).

	% Occlusion	*Fred*	*SCUD*	*T72*	*T80*	*M1a1*
Fred	10	100.0	0.0	0.0	0.0	0.0
	20	99.2	0.0	0.1	0.4	0.3
	30	95.9	0.2	0.6	1.9	1.4
	40	87.1	0.7	2.8	5.5	3.9
	50	73.2	1.6	7.1	12.1	6.0
SCUD	10	0.0	100.0	0.0	0.0	0.0
	20	0.0	99.7	0.2	0.1	0.0
	30	0.9	97.3	1.2	0.4	0.3
	40	3.1	88.8	4.9	1.9	1.3
	50	5.6	77.9	11.9	2.7	1.9
T72	10	0.0	0.0	100.0	0.0	0.0
	20	0.4	0.2	99.2	0.1	0.2
	30	2.4	0.5	95.3	1.1	0.6
	40	9.1	2.1	82.5	3.8	2.4
	50	16.8	5.2	65.9	6.8	5.4
T80	10	0.0	0.0	0.0	100.0	0.0
	20	1.2	0.0	0.1	98.6	0.1
	30	6.9	0.0	0.6	91.1	1.4
	40	21.5	0.1	1.6	72.6	4.2
	50	37.4	0.8	3.1	50.9	7.8
M1a1	10	0.0	0.0	0.0	0.0	100.0
	20	1.6	0.0	0.1	0.3	98.0
	30	8.5	0.2	0.7	2.9	87.8
	40	22.5	0.8	2.0	8.5	66.1
	50	36.9	1.1	5.2	13.8	42.9

References

[1] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE,* Vol. 77(2), pp. 257-285, 1989.

[2] L. R. Rabiner and B. H. Juang, "An introduction to Hidden Markov Models", *IEEE ASSP Magazine,* Vol. 3(1), pp. 4-16, 1986.

[3] L. R. Rabiner, B. H. Juang, S. E. Levinson and M. M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities", *AT&T Technical Journal,* Vol. 64(6), pp. 1211-1233, 1985.

[4] O. E. Agazzi and S. S. Kuo, "Pseudo Two-Dimensional Hidden Markov Models for Document Recognition", *AT & T Technical Journal,* Vol. 72(5), pp. 60-72, 1993.

[5] O. E. Agazzi and S. S. Kuo, "Hidden Markov Model based optical character recognition in the presence of deterministic transformations", *Pattern Recognition,* Vol. 26(12), pp. 1813-1826, November, 1993.

[6] B. Bhanu, G. Jones, J. Ahn, M. Li and J. Yi, "Recognition of Articulated Objects in SAR Images", *In Proceedings DARPA Image Understanding Workshop,* pp. 1237-1250, 1996.

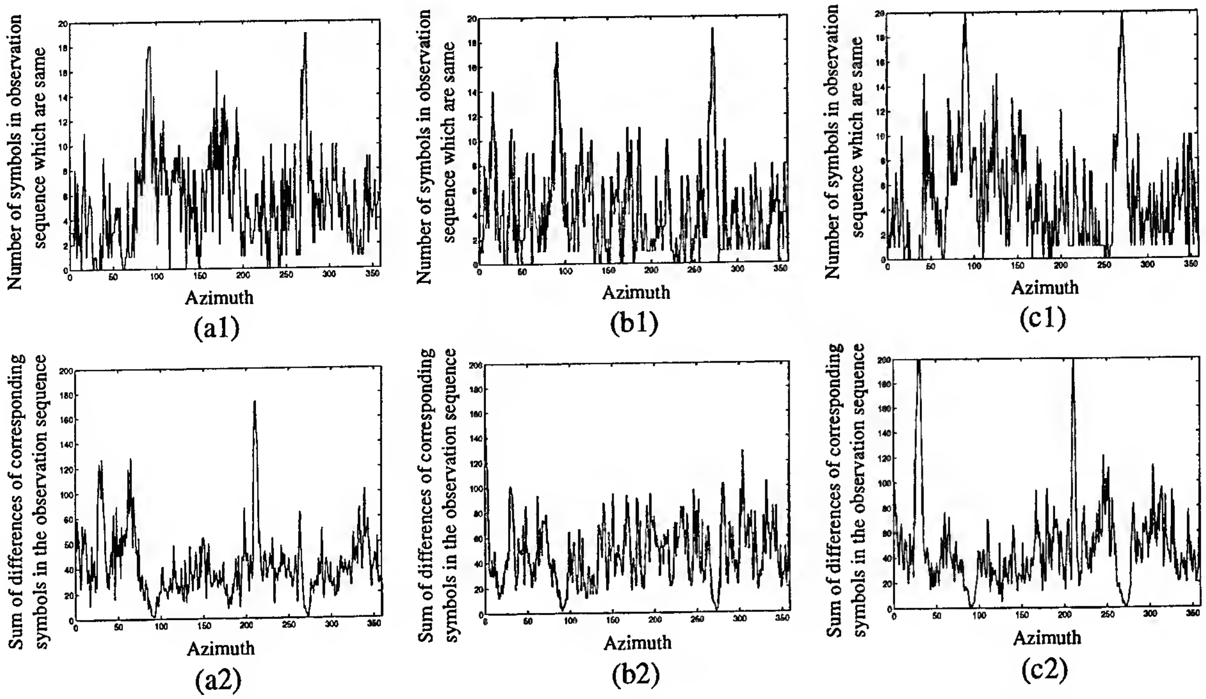


Figure 12: Comparison of observation sequence O_1 extracted from three sets of images for T72 tank. (a1, a2) 0° vs. 60°, (b1, b2) 0° vs. 90°, (c1, c2) 60° vs. 90°.

Table 4: Indexing results for articulated objects (id only).

Training T72	Testing T72	Indexing (top 5) performance
Turret at 0° and 60° 720 images	Turret at 90° 360 images	95.2
Turret at 0° and 90° 720 images	Turret at 60° 360 images	94.0
Turret at 60° and 90° 720 images	Turret at 0° 360 images	94.6
Average Performance		94.6

[7] K. H. Fielding and D. W. Ruck, "Spatio-Temporal Pattern Recognition Using Hidden Markov Models", *IEEE Trans. on AES*, Vol. 31(4), pp. 1292-1300, 1995.

[8] R. R. Rao and R. M. Mersereau, "On Merging Hidden Markov Models with Deformable Templates", *ICIP Proc.*, pp. 556-559, 1995.

[9] L. M. Novak, G. J. Owirka and C. M. Netishen, "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System", *The Lincoln Laboratory Journal*, Vol. 6(1), pp. 11-24, 1993.

[10] J.H. Yi, B. Bhanu and M. Li, "Target indexing in SAR images using scattering centers and Hausdorff distance", *Pattern Recognition Letters*, Vol. 17, pp. 1191-1198, September 1996.

[11] K. Ikeuchi, T. Shakunaga, M. D. Wheeler, and T. Yamazaki, "Invariant Histograms and Deformable Template Matching for SAR Target Recognition", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 100-105, 1996.

[12] D. J. Andersh, S. W. Lee, H. Ling and C. L. Yu, "XPATCH: A High Frequency Electromagnetic Scattering Prediction Code Using Shooting and Bouncing Ray", *Proceedings of Ground Target Modeling and Validation Conference*, pp. 498-507, 1994.

[13] K. F. Lee, *Automatic Speech Recognition - The Development of the SPHINX System,* Kluwer Academic Publishers, 1989.

[14] J. P. Fitch, *Synthetic Aperture Radar,* Springer-Verlag, 1988.

[15] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles,* Addison-Wesley, 1974.

SCALE-BASED ROBUST IMAGE SEGMENTATION

A. Kim, I. Pollak, H. Krim and A.S. Willsky
Stochastic Systems Group,
LIDS, MIT, Cambridge MA 02139

Abstract

Image segmentation represents an essential step in the early stages of an Automatic Target Recognition system. We propose two robust approaches fundamentally based on scale information inherent to a given imagery. The first approach is parametric in that the scale evolution of an image is statistically captured by a model which is in turn utilized to classify the pixels in the image. The second, on the other hand, nonlinearly evolves the image along some specific characteristic to unravel and delineate the various comprising entities in it.

1 Introduction

The growing interest in Automatic Target Recognition (ATR) is primarily due to its importance in many applications ranging from manufacturing to remote sensing and surface surveillance. Analysis and classification of various entities of an image are often of great interest, and its partitioning into a set of homogeneous regions or objects is thus of importance. The mere size of some imagery constitutes a major hurdle. A case in point is Synthetic Aperture Radar (SAR) imagery for which terrain coverage rates are very high (in excess of 1 km^2/s) and which with daunting computational demands, make algorithmic efficiency of central importance.

A SAR image reflects a coherent integration of scatterer returns (i.e. reflectivity characteristics) within a resolution cell. The number of scatterers which coherently sum up within a cell will vary with the resolution. This leads to a variation in the underlying statistics of the image. Natural clutter for example, tends to consist of a large number of equivalued scatterers, and this in contrast to the man-made one, mostly comprising a few prominent scatterers. It is precisely this type of statistical characteristic that we are interested in capturing and subsequently using as the basis for our classification of various terrain types in SAR. In addressing such a problem, we have two basic approaches, the first aims at characterizing the statistics of the image evolution in scale and ultimately using them in the pixel classification; the second attempts to diffuse "non-complying" observations via a nonlinear evolution to make it converge to specific desired domains of attraction.

The goal of this paper is to address a fundamental problem arising in ATR, namely the robust and efficient segmentation of imagery into homogeneous regions, and in presence of perhaps severe noise. To proceed, we introduce in the next section a multiscale stochastic modeling framework which affords one to capture the evolution in scale of a given image process and to statistically characterize regions of interest. Two algorithms based on this framework will be described and shown to lead to efficient and accurate segmentation. In Section 4, we present our second method based on a variational approach, and which consists of nonlinearly evolving a given image along some specific geometric constraints built around an energy functional. Finally, in Section 5, we provide a number of examples substantiating the proposed algorithms using real data imagery and accurate estimates of boundaries.

[0] This report describes research supported in part by DARPA under contract FA49620-93-1-0604.

2 Stochastic Modeling

2.1 Multiscale Stochastic Models

In this subsection, we describe a general multiscale modeling framework e.g. [2] and its adaptation to classification/identification problems. Under this framework, a multiscale process is mapped onto nodes of a *qth* order *tree*, where q depends upon how the process progresses in scale.

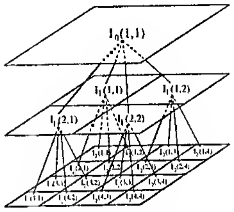


Fig. 1: Multiresolution tree.

As illustrated in Fig. 1, a *qth* order tree is a connected graph in which each *node*, starting at some root node, branches off to q child nodes. As described above, the appropriate representation for a multiscale SAR image sequence is $q = 4$, a quadtree. Each level of the tree (i.e., distance in nodes from the root node) can be viewed as a distinct scale representation of a random process, with the resolutions proceeding from coarse to fine as the tree is traversed from top to bottom (root node to terminal nodes). A coarse-scale shift operator, $\bar{\gamma}$ is defined to reference the *parent* of node s, just as the shift operator z allows referencing of previous states in discrete time-series. The state elements at these nodes may be modeled by the coarse-to-fine recursion

$$\mathbf{x}(\mathrm{s}) \;=\; \mathbf{A}(\mathrm{s})\mathbf{x}(\mathrm{s}\bar{\gamma}) \;+\; \mathbf{B}(\mathrm{s})\mathbf{w}(\mathrm{s}). \qquad (1)$$

In this recursion, $\mathbf{A}(s)$ and $\mathbf{B}(s)$ are matrices of appropriate dimension and the term $\mathbf{w}(\mathrm{s})$ represents white driving noise. The matrix $\mathbf{A}(s)$ captures the deterministic progression from node $s\bar{\gamma}$ to node s, i.e., the part of $\mathbf{x}(s)$ predictable from $\mathbf{x}(s\bar{\gamma})$, while the term $\mathbf{B}(s)\mathbf{w}(s)$ represents the unpredictable component added in the progression. An attractive feature of this framework is the efficiency it provides for signal processing algorithms. This stems from the *Markov* property of the multiscale model class, which states that, conditioned on the value of the state at any node s, the processes defined on each of the distinct subtrees extending away from node s are mutually independent.

For pixel classification purposes, a multiscale model can be constructed for each specific homogeneous class. To specify each model, it is necessary to determine the appropriate coefficients in the matrices, $\mathbf{A}(s)$ and $\mathbf{B}(s)$, and the statistical properties of the driving noise, $\mathbf{w}(s)$. Once the models have been specified, a likelihood ratio test can be derived to segment the imagery into the clutter classes.

For a binary classification problem (i.e. requiring a binary hypothesis test) each pixel in the image corresponds to one of two hypotheses: the pixel is part of some texture (H_g) or another (H_f). By exploiting the Markov property associated with the multiscale models for imagery, the log-likelihood ratio test for classifying each pixel can be written as,

$$\begin{aligned} \ell \;&= \textstyle\sum_s \log\left[p_{\mathbf{x}(s)|\beta_g}(\mathbf{X}(s) \mid \beta_g)\right] - \\ &\textstyle\sum_s \log\left[p_{\mathbf{x}(s)|\beta_f}(\mathbf{X}(s) \mid \beta_f)\right], \qquad (2) \end{aligned}$$

where $\beta_{(\cdot)} \;=\; (\mathbf{X}(s\bar{\gamma}), H_{(\cdot)})$, and $p_{\mathbf{x}(s)|\beta_g}$ and $p_{\mathbf{x}(s)|\beta_f}$ are the conditional distributions for the two hypothesized models. In the next subsection, we will show that this likelihood test can be efficiently computed in terms of the distributions for $\mathbf{w}(s)$ under the two hypotheses.

2.2 Scale-Autoregressive SAR Model

In this paper, we focus on a specific class of multiscale models, namely scale-autoregressive models [2] of the form

$$\begin{aligned} I(s) \;=\; & a_1(s)I(s\bar{\gamma}) + a_2(s)I(s\bar{\gamma}^2) + \ldots + \\ & a_R(s)I(s\bar{\gamma}^R) + w(s),\; a_i(s) \in I\!R \quad (3) \end{aligned}$$

where $w(s)$ is white driving noise and "s" is a three-tuple vector denoting scale (or resolution level), and spatial coordinates (q, n). For homogeneous regions of texture, the prediction coefficients (the $a_i(s)$ in (3)) are constant with respect to image location for any given scale. That is, the coefficients, $a_1(s), \ldots, a_R(s)$, depend only on the scale of node s (denoted by $m(s)$), and thus will be denoted by

$a_{1,m(s)}, \cdots, a_{R,m(s)}$. Furthermore, the probability distribution for $w(s)$ depends only on $m(s)$. Thus, specifying both the scale-regression coefficients and the probability distribution for $w(s)$ at each scale completely specify the model.

Following the procedure of state augmentation used in converting autoregressive time series models to state space models, we associate to each node s a R-dimensional vector of pixel values, where R is the order of the regression in (3). The components of this vector correspond to the SAR image pixel associated with node s and its first $R-1$ ancestors. Specifically, we define

$$\mathrm{x}(s) = \begin{bmatrix} I(s) & I(s\bar{\gamma}) & \cdots & I(s\bar{\gamma}^{R-1}) \end{bmatrix}^T. \quad (4)$$

Thus, for a model of the form (3) or equivalently (1), ℓ in (2) can be calculated using

$$p_{\mathrm{x}(s)|\mathrm{x}(s\bar{\gamma})}(\mathbf{X}(s) \mid \mathbf{X}(s\bar{\gamma})) \;=\; p_{w(s)}(W(s)) (5)$$

with $W(s)$ being the vector of residuals at scale "s". The identification of the model for each clutter class can thus be obtained for each scale m by a standard least-squares minimization,

$$\boldsymbol{\alpha}_m = \arg \min_{\boldsymbol{\alpha}_m \in I\!R^R} \left\{ \sum_{\{s \,|\, m(s)=m\}} [I(s) - a_{1,m}I(s\bar{\gamma}) - \ldots - a_{R,m}I(s\bar{\gamma}^R)]^2 \right\} (6)$$

where

$$\boldsymbol{\alpha}_m = \begin{bmatrix} a_{1,m} \; a_{2,m} \; \cdots \; a_{R,m} \end{bmatrix}^T,$$

with R being the regression model order. For most of the results presented in Section 5 a third order regression ($R=3$) for both the grass and the forest models was chosen. To obtain a statistical characterization of the prediction error residuals (the $w(s)$ in (3)) of the model at scale m, we evaluate the residuals in predicting scale m of a homogeneous test region. In particular, we use the $\boldsymbol{\alpha}_{m,\mu}$ found in (6) to evaluate all $\{\mathbf{w}(s)|m(s)=m\}$ as indicated by Eq. (5) with μ specifying "grass" or "forest".

3 Model-Based Classification

3.1 Residual-Based Classification

While we could conceivably postulate a spatial random field model for each windowed clutter category to classify its center pixel, we use to advantage the efficiency of multiscale likelihood calculation to base the classification of each individual pixel "s" on a surrounding $(2K+1) \times (2K+1)$ window $\mathcal{W}(\mathbf{s})$, where the parameter K is a nonnegative and judiciously selected integer. While a larger window provides a more accurate classification of homogeneous regions, it also increases the likelihood that the window contains a clutter boundary. Thus, keeping the window size as small as possible is also desirable. As shown in [1] in the next section, one can determine the tradeoff between classification accuracy and window size by examining the empirical distribution of ℓ over windows of various size for homogeneous regions of grass and forest.

3.1.1 Statistical Hypotheses Test

Whenever a clutter boundary is present within a test window the validity of the center pixel classification is questionable. This effect results in a classification bias near boundaries. To address this problem, we devise a method to detect the proximity of grass-forest boundaries and subsequently utilize a procedure to refine the classification. Terrain boundary proximity is detected via a simple modification of the decision made based on the test statistic ℓ. Specifically, rather than comparing ℓ to a single threshold to decide on a grass-or-forest classification, we compare ℓ to two thresholds a and b as shown in Fig. 2, and where we include a defer decision.

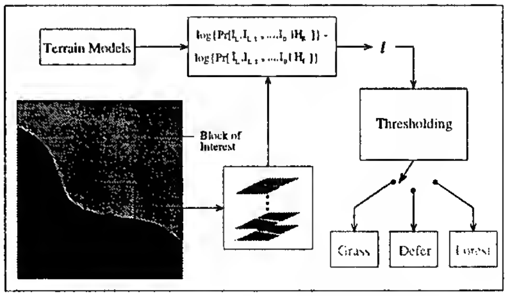

Terrain Models
log{Pr[I_L,I_{L-1},...,I_0|H_g]} -
log{Pr[I_L,I_{L-1},...,I_0|H_f]}
ℓ
Block of Interest
Thresholding
Grass
Defer
Forest

Fig. 2: Initial pixel classification.

This is tantamount to refining the decision procedure by considering smaller windows within the original window until a majority rule lifts the ambiguity,

$\mathrm{H_g} : \ell > a$ Classify as Grass,

$a > \ell > b$ Defer decision
(possible boundary presence),

$\mathrm{H_f} : \ell < b$ Classify as Forest.

3.2 Parameter-Based Classification

An alternative approach to classifying a clutter type is to evaluate an ℓ^2 distance between the computed model parameters for a window region $\mathcal{W}(\mathbf{s})$ and those of a template homogeneous region [3]. For each pixel "s" (or equivalently tree node) a model for a corresponding surrounding window $\mathcal{W}(s)$ is computed at several scales giving rise to what we refer to as an evolution vector statistic,

$$\mathbf{y}(\mathbf{s}) = [\boldsymbol{\alpha}_{m(s)}, \boldsymbol{\alpha}_{m(s)-1}, \cdots, \boldsymbol{\alpha}_1], \qquad (7)$$

where "s" will sweep all of the three-tuple vectors $[m(s), q, n]$ at the $m(s)^{th}$ resolution with $m(s) = 1, \cdots, L-1$ and L the cardinality of the considered set of images. We should note at this point that in this approach, the DC component in the $\boldsymbol{\alpha}_{(\cdot)}$ is included. The order of the regression associated with modeling $\mathcal{W}([m(s), q, n])$ from its ancestors will vary with the level $m(s)$ as defined by the function

$$O_{m(s)} = \max(R, m(s))$$

The regression vector $\boldsymbol{\alpha}(m(s))$ provides a statistically optimal description of the linear dependency of $\mathcal{W}(\mathbf{s})$ on $\{\mathcal{W}(s\gamma^j)\}_{j=1}^{O_{m(s)}}$, and $\mathbf{y}(\mathbf{s})$ thus being a measure of the scale evolution behavior of the windowed region.

Here again the size of the window used to compute the evolution vector, is the result of a tradeoff between *modeling consistency* and *local accuracy.*

3.2.1 Statistical Classification

A characterization of the evolution vector $\mathbf{y}(s)$ is necessary to carry out a statistically meaningful classification of various types of terrain. Specifically, a BHT is applied to the evolution $\mathbf{y}(\mathbf{s})$ in order to classify node $\mathbf{s}$ as a member of either a region of grass or of forest. These hypotheses are respectively designated as hypotheses H_g and H_f. The classification of pixel $m(s)$ will depend only on $\mathbf{y}(\mathbf{s})$ and the predetermined likelihoods $p\,(\mathbf{y}(\mathbf{s})|H_g)$ and $p\,(\mathbf{y}(\mathbf{s})|H_f)$.

To carry out a statistically significant hypothesis test, we need to specify the conditional probability density for $\mathbf{y}$ under each hypothesis. To do this, we extensively examined the distribution of the evolution vectors obtained from a large homogeneous region of the corresponding terrain. For both grass and forest terrain, it turns out that each component in $\mathbf{y}$ approximately has a Gaussian distribution. We consequently make the approximation that $p\,(\mathbf{y}|H_g)$ and $p\,(\mathbf{y}|H_f)$ are N-variate Gaussian densities. They are then completely specified by their mean vectors $\mathbf{m}_g$ and $\mathbf{m}_f$ and their covariance matrices K_g and K_f all of which are calculated from the training data for each hypothesis. A maximum likelihood (ML) detector is then used to classify each pixel (using the ML detector assumes equal a priori probabilities for each hypothesis and a cost function that penalizes all misclassifications equally). In implementing the ML detector, a threshold η is calculated from the likelihoods and used in the classification of each pixel through its comparison to a sufficient statistic derived from the evolution vector. Because $p\,(\mathbf{y}|H_g)$ and $p\,(\mathbf{y}|H_f)$ are assumed to be Gaussian, it is straightforward to compute the threshold η and the sufficient statistic $\ell'\,(\mathbf{y})$ for each evolution vector as

$$\eta = \log \frac{|K_g|}{|K_f|}, \qquad (8)$$

and

$$\begin{aligned} \ell'\,(\mathbf{y}) \;\; &= \;\; (\mathbf{y} - \mathbf{m}_f)^T K_f^{-1} (\mathbf{y} - \mathbf{m}_f) - \\ & \quad\;\; (\mathbf{y} - \mathbf{m}_g)^T K_g^{-1} (\mathbf{y} - \mathbf{m}_g) . \qquad (9) \end{aligned}$$

The classification of $\mathbf{I}(\mathbf{s})$, denoted as $\mathbf{C}(\mathbf{s})$, is then given by

$$\mathbf{C}_\mathrm{s} = \begin{cases} H_g & \text{if } \eta \geq \ell\left(\mathbf{y}_{[q,n]}\right) \\ H_f & \text{if } \eta < \ell\left(\mathbf{y}_{[q,n]}\right). \end{cases} \qquad (10)$$

The construction of the evolution $\mathbf{y}$ and subsequent application of a BHT for each $s \in$

$\{s \mid m(s) = m\}$ thus provide a segmentation of $\mathbf{I}_m$. Instead of independently generating a segmentation for each image resolution for all $s \in \{s \mid m(s) < L\}$, $\mathbf{C}(\mathbf{s})$ can be obtained by comparing η to the average of the sufficient statistics of nodes in $\mathbf{I}_L$ which have $\mathbf{s}$ as an ancestor, i.e.

$$\mathbf{C}_{[1,m,n]} = \begin{cases} H_g & \text{if } \left(2^{m(s)-1}\right)^2 \eta \geq \sum_{\mathbf{s}} \ell(\mathbf{y}(\mathbf{s})) \\ H_f & \text{if } \left(2^{m(s)-1}\right)^2 \eta < \sum_{\mathbf{s}} \ell(\mathbf{y}(\mathbf{s})). \end{cases} \tag{11}$$

Although this does not yield a sufficient statistic for $\mathbf{I}(\mathbf{s})$, doing so is computationally more efficient than calculating a sufficient statistic as in Eq. (9) for each node at every level in the quadtree. Note that the segmentation technique described here, could easily be generalized to a larger number of terrain types.

4 Stabilized Inverse Diffusion Equations (SIDEs)

The previous two techniques rely on modeling a linear evolution of the observed imagery with an ultimate goal of pixel classification. In this section, we instead carry out a nonlinear evolution which is driven by prescribed geometric features underlying the process/imagery, and study its progression.

Towards that end, we introduce a discontinuous force function, resulting in a system of equations that has discontinuous right-hand side (RHS). As shown below, the objective is to drive an evolution trajectory onto a lower-dimensional surface which clearly has value in image analysis, and in particular in image segmentation. Segmenting a signal or image, represented as a high-dimensional vector $\mathbf{I}$, consists of evolving it so that it is driven onto a comparatively low-dimensional subspace, which corresponds to a segmentation of the signal or image domain into a small number of regions.

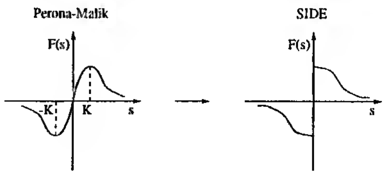


Fig 3: Force function for a SIDE.

The type of force function of interest to us here is illustrated in Fig. 3(right). More precisely, we wish to consider force functions $F(v)$ which, in addition to driving the following evolutions,

$$\begin{aligned} \dot{\mathbf{I}}(s) &= F(\mathbf{I})(s), \\ \mathbf{I}(0) &= \mathbf{I}_0, \end{aligned} \tag{12}$$

where "s" is now a continuous scale and a " dot " denotes differentiation with respect to it, satisfy the following properties:

$$\begin{aligned} F'(v) \leq 0 \quad & \text{for}\, v \neq 0, \text{ and } F(0^+) > 0, \\ & F(v_1) = F(v_2) \Leftrightarrow v_1 = v_2. \end{aligned} \tag{13}$$

Contrasting this form of a force function to the Perona-Malik function [4] in Fig. 3 (left), we see that in a sense, one can view the discontinuous force function as a limiting form of the continuous force function in Fig. 3 (left). We, however, need a special definition of how the trajectory of our evolution proceeds at the discontinuity points $F(0^+) \neq F(0^-)$. For this definition to be useful, the resulting evolution must satisfy well-posedness properties: the existence and uniqueness of solutions, as well as stability of solutions with respect to the initial data. We address the issue of well-posedness and other properties in [5].

Considering the evolution (12) with $F(v)$ as in Fig. 3(right) in a SIDE, notice that the RHS of (12) has a discontinuity at a point $I_{(\cdot)}$ if and only if $I_i = I_{i+1}$ for some i between 1 and $N-1$. It is when a trajectory reaches such a point $I_{(\cdot)}$ that we need the following definition:

$$\dot{I}_i = \dot{I}_{i+1} = \frac{1}{2}((F(I_{i+2} - I_{i+1}) - F(I_i - I_{i-1})). \tag{14}$$

In other words, the two observations are simply merged into a single one, resulting in Eq. 14 for $n = i$ and $n = i + 1$ (the differential equations for $n \neq i, i + 1$ do not change.).

Similarly, if q consecutive observations become equal, they are merged into one which is weighted by $1/q$ [5]. The evolution can then naturally be thought of as a sequence of stages: during each stage, the right-hand side of (12) is continuous. Once the solution hits a discontinuity surface of the right-hand side, the state reduction and re-assignment of q_{n_i}'s, described above, takes place. The solution then proceeds according to the modified equation until the next discontinuity surface, etc.

Notice that such an evolution automatically produces a multiscale segmentation of the original signal if we view each compound observation as a region of the signal. The algorithm may be be summarized as follows:

1. Start with the trivial initial segmentation: each sample is a distinct region.
2. Evolve (12) until the values in two or more neighboring regions become equal.
3. Merge the neighboring regions whose values are equal.
4. Go to step 2.

The same algorithm can be used for 2-D images, which is immediate upon re-writing Eq. (12) and an example is provided next.

5 Experiments

A number of segmentation experiments have been carried out on SAR imagery using the three techniques described above. Fig. 4(b) shows a segmentation as a result of a systematic pixel classification as described in Technique 1. A brute force as well as a refined segmentations are shown, and the hand drawn of the eye-balled boundary is shown in white. In Fig. 4(c), the model-based segmentation is illustrated and in Fig. 5 the nonlinear evolution-based segmentation is shown which clearly is perhaps the most promising, albeit at a slightly higher computational cost.

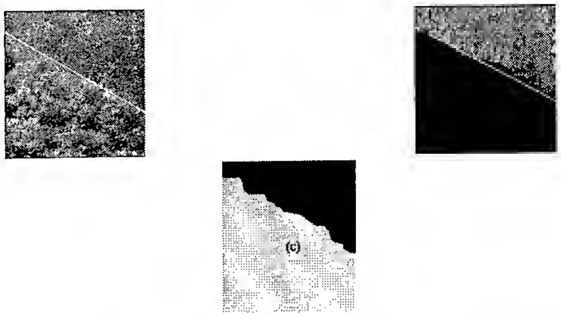


Fig. 4: Residual and Model-Based Segmentation (b and c).

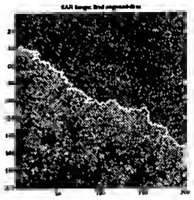


Fig. 5: Segmentation by Nonlinear Diffusion.

References

[1] C. Fosgate, H. Krim, W. Irving, W. Karl, and A. Willsky. Multiscale segmentation and anomaly enhancement of SAR imagery. *IEEE Trans. on Im. Proc.*, 6(1):7–20, Jan. 96.

[2] W.W. Irving. Multiresolution approach to discriminating targets from clutter in SAR imagery. In *Proc. of SPIE Symp.*, Orlando, FL., 1995.

[3] A. Kim and H. Krim. Hierarchical stochastic modeling of SAR imagery for segmentation/compression. *submitted to special issue of IEEE Trans. on SP*, 1996.

[4] P. Perona and J. Malik. Scale-space and edge detection using anistropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12, 1990.

[5] I. Pollak, A.S. Willsky, and H. Krim. Image segmentation and edge enhancement with stabilized inverse diffusion equations. *To be submitted to IEEE Trans. on Image Processing.*

Invariants for the Recognition of Articulated and Occluded Objects in SAR Images

Grinnell Jones III and Bir Bhanu*
College of Engineering, University of California, Riverside, CA 92521
{grinnell, bhanu} @constitution.ucr.edu
URL: http://constitution.ucr.edu

Abstract

Articulation invariant features are found (and quantified) in Synthetic Aperture Radar (SAR) images of military vehicles. They are used in the development of a SAR recognition engine that successfully identified articulated objects based on non-articulated recognition models. The engine also achieves robust recognition performance with mostly spurious data from noise or highly occluded objects. Performance results are related to the percent of invariant or unoccluded features.

1 Introduction

1.1 Problem Definition and Scope

Automated object recognition in SAR imagery is a significant problem because recent developments in image collection platforms will soon produce far more imagery (terabytes per day per aircraft) than the declining ranks of 2000 image analysts are capable of handling [10]. The specific challenges of this research are to address the need for automated recognition of military vehicles that can be in articulated configurations (such as: tanks, where the turret can rotate and the SCUD missile launcher, where the missile can erect) and can be partially hidden. Previous recognition methods involving template matching [11] are not useful in these cases, because articulation or occlusion changes global features like the object outline and major axis. In this paper the problem scope is the recognition subsystem itself, starting with SAR chips of target vehicles and ending with the vehicle identification. Because the very high resolution SAR target chips are not openly available, the U.S. Air Force provided the XPATCH high frequency radar signature prediction code [1], which is used to construct 4320 target chips for this research.

1.2 Overview of Approach

Our approach to object identification is specifically designed for SAR. The peaks (local maxima) in radar return are related to the physical geometry of the object. The relative locations of these scattering centers are independent of translation and serve as distinguishing features. The specular radar return varies greatly with the uncontrolled target orientation (azimuth). This azimuthal variance is captured by using 360 azimuth models. (The radar depression angle to the target is controllable, or known, and it is fixed at 15^o for this study). Useful articulation invariants are found, which permit building non-articulated recognition models and using them to successfully recognize articulated targets. The SAR recognition engine, Figure 1, has an off-line model construction phase and an on-line recognition process. The recognition model is a look-up table that relates the relative distances among the scattering centers (in the radar range and cross range directions) to object type and azimuth. The recognition process is an efficient search for *positive evidence*, using relative locations of scattering centers to access the look-up table and generate votes for the appropriate object (and azimuth).

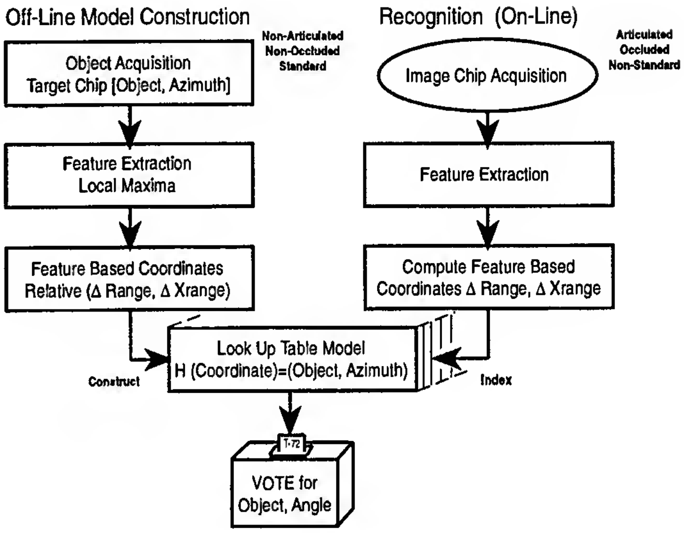


Figure 1: SAR recognition engine architecture.

1.3 Related Work and Our Contribution

A comparison of this approach, for the articulated and occluded object recognition problems in SAR, with related work is given in Table 1. Our approach is designed specifically for SAR, but is related to geometric hashing [8]. Scattering center relative positions are used as SAR recognition features. Template matching

*Supported in part by grants MDA972-93-1-0010 and DAAH04-95-1-0488; the contents and information do not reflect positions or policies of the U.S. Government.

Table 1: Related work comparisons.

Area	Related Approach	This Approach
Indexing:	Geometric Hashing [8]:	SAR specific:
• transformation	translation	translation
	scale	scale fixed in SAR
	rotation	azimuth models
• bin type/size	real/varies	integer/fixed
SAR Recognition:	Template Matching [11]:	Peak Locations:
• azimuth models	12 blobs	360 constellations
• reference frame	global	local
Occlusion:	Invariant Histogram [7]:	Recognition Engine:
• azimuth models	36 (10° apart)	360 (1° apart)
• bin size	2-4 feet	6 inches
• near neighbor?	yes	no
• search	exhaustive (L_1 norm)	voting
Articulation	Constraint Models [2] [6]	Invariants

[11] is not suitable for recognizing articulated and occluded objects since there will be a combinatorial explosion of the number of templates with varying articulations and occlusions. The SAR recognition engine presented here has sufficient precision to perform both indexing and matching functions, while the invariant histogram technique (that has been applied to recognize occluded objects [7]) has limited performance, is only capable of indexing object models and requires a seperate template matching step. Constrained models of parts and joint articulation, used in optical images [2] [6], are not appropriate for the relatively low resolution, non-literal nature and complex part interactions of SAR images; which are handled by using articulation invariants as discussed in this paper. The major contributions of this paper are:

1. Identifies and quantifies articulation invariants.
2. Demonstrates a SAR recognition engine with robust performance for articulated and occluded objects.
3. Relates performance with invariance of features.
4. Quantifies azimuthal variance.

2 SAR Scattering Centers

The relative locations of peaks in the radar return are characteristic features that are related to the geometry of the object. The typical detailed edge and straight line features of man-made objects in the visual world, do not have good counterparts in SAR images for sub-components of vehicle sized objects at six inch to two foot resolution. The amplitude map, Figure 2, of a typical SAR target (SCUD launcher with missile erect, 18° azimuth, 15° depression) at six inch resolution shows a wealth of peaks corresponding to scattering centers and has no obvious lines or edges within the boundary of the vehicle. The 4320 target chips for the T72, T80, M1a1, FRED tanks and the SCUD missile launcher have a range of 52 to 284 local peaks. The locations of the peaks are related to the aspect and detailed geometry of the object. For example, for the T72 tank model, the strongest returns (that persist for 20° or more in azimuth span) are from four trihedral corners on the upper rear deck of the tank hull [3]. Figure 3 shows target geometries (model sizes in increasing order: T80, M1a1,T72 and SCUD launcher). The tank turret angle is measured counter-clockwise from the hull forward centerline.

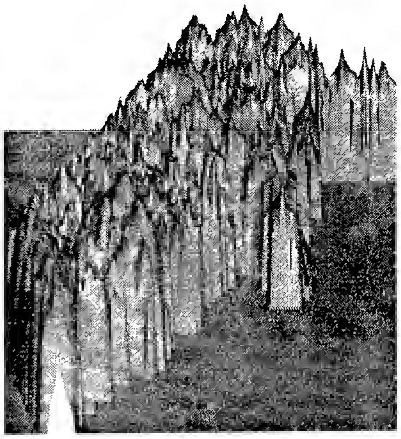
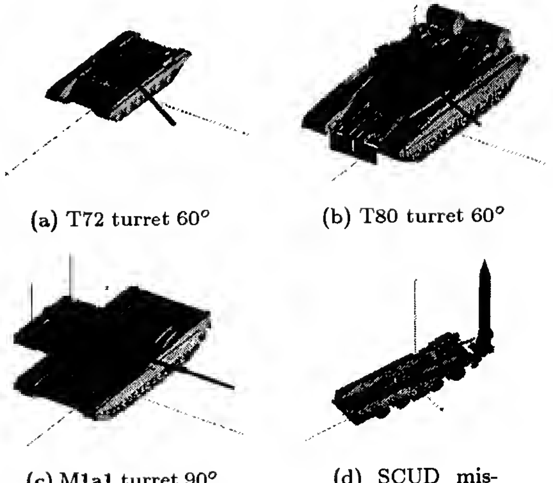


Figure 2: Example SAR image amplitude map.

2.1 Azimuthal Variance

The typical rigid body rotational transformations for viewing objects in the visual world do not apply much for the specular radar reflections of SAR images, because *significant* numbers of features *do not* typically persist over a few degrees of rotation. Averaging the results for 360 azimuths of the T72 tank, only about one-third of the 50 strongest scattering center locations (in object centered coordinates) remain unchanged (i.e. within an error radius of 1/2 pixel) for a 1° azimuth change (see Figure 4) and less than 5% persist for 10°. These are significantly less than the one foot resolution ISAR results of Dudgeon [5], whose



(a) T72 turret 60°

(b) T80 turret 60°

(c) M1a1 turret 90°

(d) SCUD missile launcher

Figure 3: Articulated objects (not to scale).

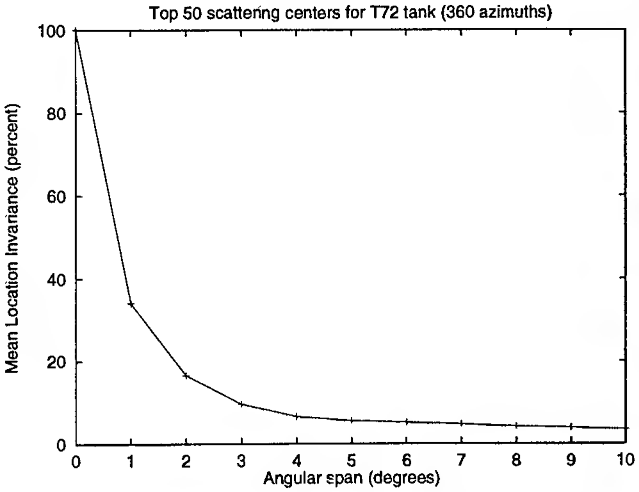
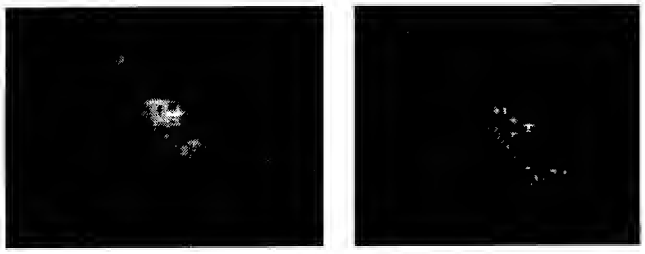


Figure 4: T72 azimuthal invariance.



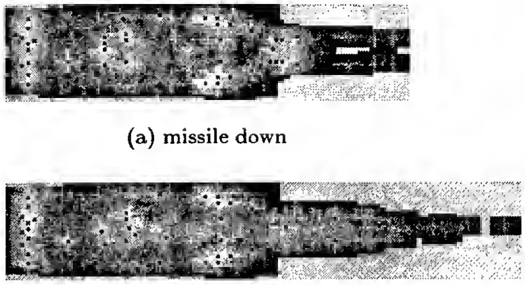
(a) two foot (b) six inch

Figure 5: SAR image resolution examples.

definition of 'persistence' allowed scintillation (i.e., a point was required to be present/absent for 2 consecutive angles, 1^o apart, to appear/disappear, thus a feature point would be 'persistent' if it appears and then disappears in images separated by 1^o). Because of the presence of azimuthal variation and the goal of recognizing articulated and occluded objects, in this research we use 360 azimuth models (1^o intervals), in contrast to others who have used 6^o [9] and 10^o [7] intervals and 12 models [11].

2.2 Image Resolution

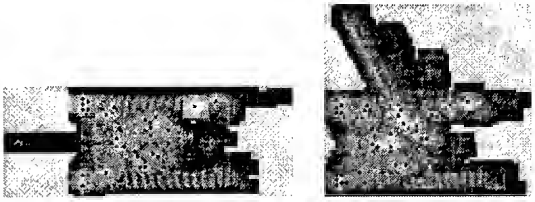
A SAR image is formed by collecting backscattered field over a frequency band and over an angular span of incident directions. The resolution and scale of objects are fixed by the operating parameters of the radar beam: frequency, frequency bandwidth and angular span. Six inch resolution X-band images (10.0 GHz center frequency, 1.0 GHz bandwidth, 5.6^o angular span) are used to provide a rich feature set to facilitate the task of recognizing articulated and occluded targets. This provides 16 times as many pixels as two foot resolution. The comparison of the two foot resolution target 'blobs' with the six inch resolution constellation of image points is shown in Figure 5 for the FRED tank.



(a) missile down

(b) missile up

Figure 6: SCUD launcher articulation example.



(a) straight turret (b) −60° turret

Figure 7: T72 articulation example.

3 Articulation Invariants

The existence of articulation invariants in six inch resolution SAR imagery can be seen in Figures 6 and 7, where the locations of scattering centers are indicated by the black squares. In the example of the SCUD launcher, with the radar directed (from the left in Figure 6) at the front (cab end) of the launcher, many of the details from the launcher itself are independent of whether the missile is erect or down. In the similar view of the T72 tank, many of the details from the hull are independent of the turret angle. An example of articulation invariance is shown in Figure 8, which plots the percentage of the strongest 50 scattering centers for the T72 tank that are in exactly the same location with the turret rotated 60^o as they are with the turret straight forward, for each of 360 azimuths. The mean μ, standard deviation σ and $\mu - \sigma$ values of the average percent articulation invariance (for 360 azimuths) is shown in Table 2 for the individual articulated objects and the overall average. Comparing the cases for 25 and 50 scattering centers, the mean values are similar, but the $\mu - \sigma$ values are consistantly less for the 25 scatterer cases. The smaller average articulation invariance for the M1a1 tank is expected, because the M1 tank has a comparatively much larger turret than the other tanks (see Figure 3).

4 SAR Recognition Engine

4.1 Local Reference Coordinate System and Translational Invariance

Establishing an appropriate local coordinate reference frame is critical to reliably identifying objects (based on locations of features) in SAR images of ar-

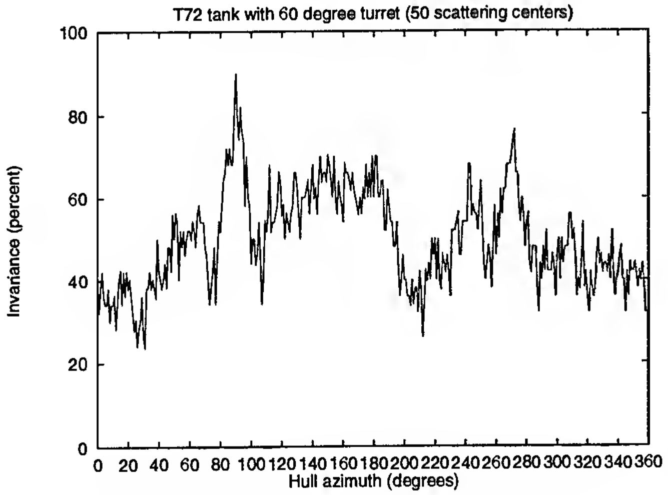


Figure 8: Articulation invariants example.

Table 2: Articulation invariance percentages.

	25 Scatterers			50 scatterers		
	μ	σ	$\mu - \sigma$	μ	σ	$\mu - \sigma$
SCUD missile up	53.26	12.40	40.86	53.67	9.79	43.88
T72 60°turret	48.76	14.88	33.88	49.94	11.71	38.23
90°turret	46.52	14.96	31.56	48.00	12.03	35.97
M1a1 60°turret	37.37	14.56	22.81	37.96	9.32	28.64
90°turret	38.33	13.83	24.50	37.66	9.06	28.60
T80 60°turret	57.70	14.69	43.01	54.17	8.02	46.15
90°turret	57.82	13.35	44.47	53.84	7.94	45.90
average	48.54	14.10	34.94	47.89	9.70	38.19

ticulated and occluded objects. The object articulation and occlusion problems require the use of a local coordinate system; global coordinates and global constraints do not work, as illustrated in Figures 6 and 7 where the center of mass and the principal axes change with articulation. In the geometry of a SAR sensor the 'squint angle', the angle between the flight path (cross-range direction) and the radar beam (range direction), can be known and fixed at 90°. Given the SAR squint angle, the image range and cross-range directions are known and any local reference point chosen, such as a scattering center location, establishes a reference coordinate system. (The scattering centers are local maxima in the radar return signal.) The relative distance and direction of the other scattering centers can be expressed in radar range and cross-range coordinates, and naturally tessellated into integer buckets that correspond to the radar range/cross-range bins. For the examples shown in Figures 6, 7 and 10 - 13 range is to the right (x axis), cross-range is up (y axis). The recognition engine takes advantage of this natural system for SAR, where a single basis point performs the translational transformation and fixes the coordinate system to a 'local' origin. For ideal data, picking the location of the strongest scattering center as the reference point is sufficient. However, for potentially corrupted data where any feature point could be spurious or missing (due to the effects of noise, articulation, occlusion, non-standard configurations, etc.), the process needs to continue with other scattering centers as the reference point to ensure a valid feature point is obtained as the origin. Although this implementation used all the scattering center locations in turn as reference points, heuristics could be applied to use fewer reference points for increased efficiency.

4.2 Recognition System Architecture

The basic system architecture, Figure 1, is an off-line model construction process and a similar on-line recognition process. The approach is based on local features and local reference coordinate systems. A systematic method is employed for constructing recognition models of objects that are not articulated, the models are stored in a look-up table and then local image features are used to index into the look-up table of models and recognize the same objects in articulated configurations. The image features used are the positions of the scattering centers (local maxima in the signal strength). The number of scattering centers used is a design parameter that is optimized based on experimental results. The positions of the scattering centers are expressed in relative distances in the known SAR range and cross-range coordinates. The model construction technique extracts these relative distances of the scattering centers from the non-articulated training data for all 360 azimuths for each target type. An example relative distance distribution for the SCUD launcher with the missile up is shown in Figure 9 (with the distances shifted by 154 range and 99 cross range to make the values non-negative). The model database is basically a table that relates these distances to object labels (target type and azimuth). The bounds of the table indices (and the shift amounts) are dictated by the relative distances between scattering centers of the largest target (the SCUD launcher with the missile up, establishes a 308 range by 198 cross-range table and the 154 range, 99 cross range shift for all the data). In the recognition phase, the relative positions of scattering centers are obtained from the articulated (or occluded) test data. These relative distances are indices into a look-up table which provides the associated object label(s) that are used to accumulate evidence for target identification. The process is repeated with different scattering centers as reference points, providing multiple 'looks' at the database. The target type and azimuth angle pair with the most 'votes' is chosen as the answer.

4.3 Algorithms and Complexity

A single test data or model (object, azimuth angle) instance i, with M features is represented by relative range (r) and cross range (c) distance distributions given by:

$$a(r,c) = \left\{ \begin{array}{l} k \\ 0 \end{array} \right. \qquad (1)$$

where $k = 1$ for the models and $\sum k \leq M(M-1)/2$ (duplicates are removed), while for the test data $k \geq 1$ and $\sum k = M(M-1)/2$ (duplicates are allowed). In the recognition phase an object, azimuth model instance i gets votes, V_i, as the intersection of the test data distance distribution and the distance distribu-

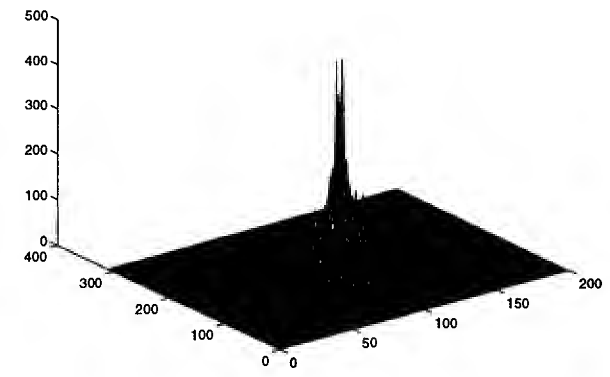


Figure 9: Relative distance distribution for SCUD launcher with the missile up.

tion a_i for model i:

$$V_i = \sum_r \sum_c a_{\text{test}}(r,c) a_i(r,c). \quad (2)$$

The rule for object identification I is given by:

$$I = \max\{V_i\} \text{ for all models } (i = 1 \text{ to } 360 \text{ N}); \quad (3)$$

where N is the number of objects. The recognition engine implements equations (1) to (3) as a look-up table and a decision rule.

The basic steps of the off-line model construction algorithm are:

1. For each of N Objects do 2:
2. For each of 360 Azimuth angles do 3,4:
3. Obtain the (row, column) locations of the strongest M peaks from image(Object, Azimuth),
4. For each of M Origins do 5:
5. For each Point from Origin + 1 to M do 6,7:
6. Find the relative Range and Cross-range positions of the Point from the Origin,
7. Add a new table entry at (Range, Cross-range) with Object, Azimuth label (add to a list if table already occupied).

The model construction algorithm complexity is 360 N M (M-1)/2, where N is the number of target types and M is the number of peaks used. The on-line recognition algorithm steps are:

1. Obtain the (row, column) locations of the strongest M peaks from test image.
2. For each of M Origins do 3:
3. For each Point from Origin + 1 to M do 4,5,6:
4. Find the relative Range and Cross-range positions of the Point from the Origin,
5. Look up the list of table entries at (Range, Cross-range),
6. Traverse the list: reading labels and incrementing Object, Azimuth label accumulators.
7. At completion, select the Object, Azimuth label with the largest accumulated total.

The recognition algorithm makes M (M - 1)/2 queries to the lookup table, where M is the number of peaks used. The only models associated with a lookup table entry are the "real" model and any other models that happen, by coincidence, to have a feature pair with the same relative distances.

5 Results

The XPATCH radar signature prediction code [1] is used to generate target chips at 360 azimuth angles (at a 15° depression angle, 90° squint angle) from the CAD models of the various objects. For 5 objects, the non-articulated set used for model construction is 1800 target chips. There are 7 sets of articulated test data (SCUD Launcher with the missile up, and the T72, M1a1, and T80 tanks with 60° and 90° turret angles) for an additional 2520 target chips. The scattering center locations are obtained at local maxima in the signal amplitude (where amplitude is greater than the surrounding eight neighbors, if equal reject unless last in raster scan order). Examples, at various azimuths, of the object geometry, SAR image and (strongest 50) scattering center locations are shown for both non-articulated and articulated cases of the T72 (Figure 10), M1a1 (Figure 11), T80 (Figure 12), and the SCUD launcher (Figure 13). (Figures 10 - 13 are not to scale and the image is displayed at 8 intensity levels, the scattering centers at 256 levels).

5.1 Articulated Objects

5.1.1 Identification and Pose (4 Object Table)

A 4 object recognition table for the SCUD missile launcher, T72, M1a1 and T80 tanks is constructed from 1440 non-articulated target chips using the model construction algorithm given above. The experimental results of 2520 trials with articulated objects for the recognition engine using 50 scattering centers and this 4 object table are shown as a confusion matrix in Table 3. The overall performance is a 93.14% probability of correct identification (PCI). The azimuth accuracy is shown in Table 4, where 'e' is an exact match and 'c' indicates a match within ±5°. The azimuth results are reported for the hull angle. In the case of the M1a1 tank, decreased azimuth accuracy results when identifications are based on the (relatively large) turret rather than the hull.

5.1.2 Number of Scattering Centers Used

The number of scattering center locations used is a design parameter that can be tuned to optimize performance of the recognition engine. For the objects and

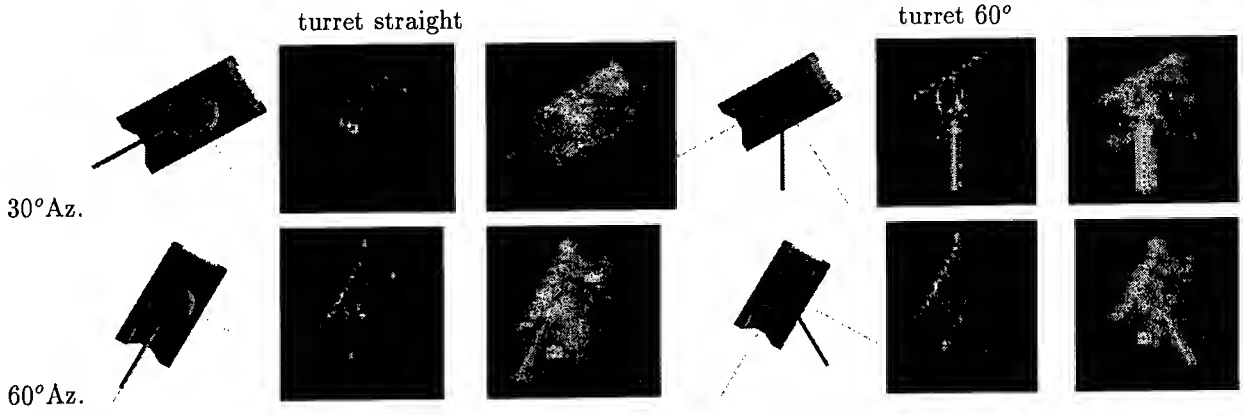


Figure 10: Examples of T72 tank geometry, SAR image and scattering centers for 30° and 60° hull azimuths.

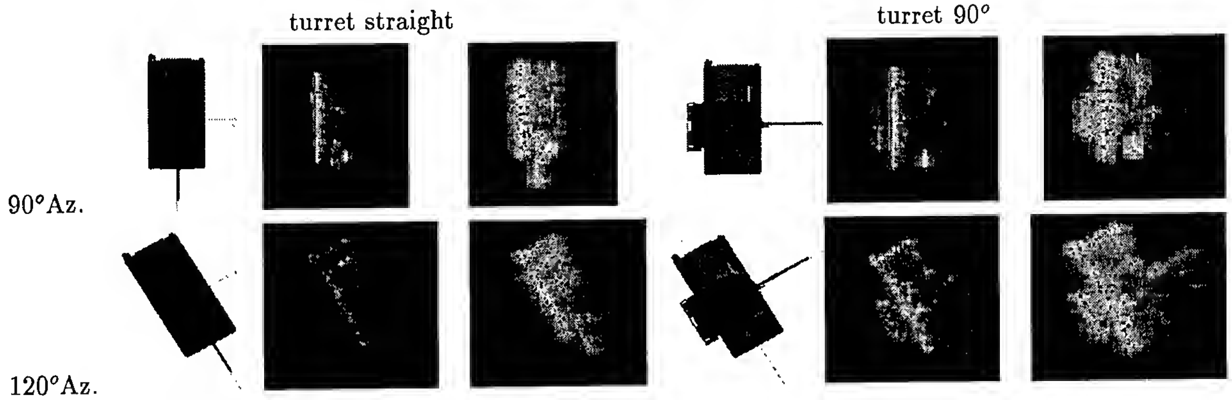


Figure 11: Examples of M1a1 tank geometry, SAR image and scattering centers for 90° and 120° hull azimuths.

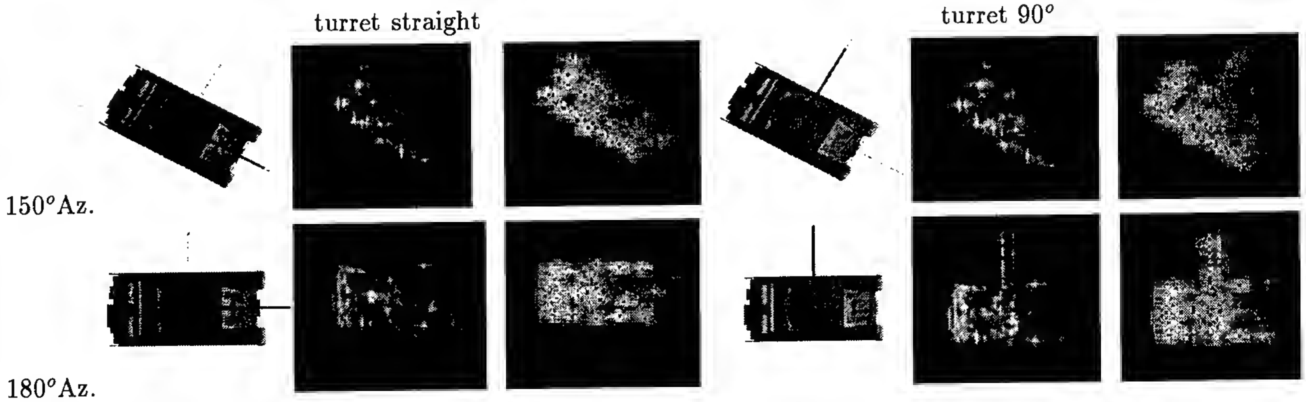


Figure 12: Examples of T80 tank geometry, SAR image and scattering centers for 150° and 180° hull azimuths.

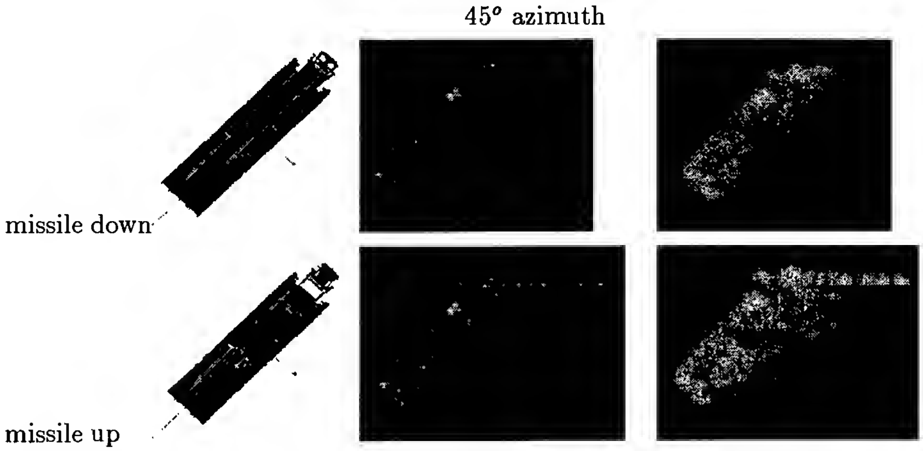


Figure 13: Examples of SCUD Launcher geometry, SAR image and scattering centers.

Table 3: Confusion matrix for articulated identification results (4 object table, 50 scattering centers).

articulated test targets	Non-articulated models			
	SCUD	T72	M1a1	T80
	down	0° turret		
SCUD missile up	360			
T72 60° turret		335	7	18
90° turret		327	8	25
M1a1 60° turret		1	300	59
90° turret		2	305	53
T80 60° turret				360
90° turret				360

Table 4: Confusion matrix for articulated pose accuracy results (e = exact pose, c = pose within ±5°).

articulated test targets	Non-articulated models			
	SCUD	T72	M1a1	T80
	down	0° turret		
SCUD missile up	360e			
T72 60° turret		333e	1c	2c
90° turret		323e		
M1a1 60° turret			261c, 254e	4c, 1e
90° turret		2c	272c, 261e	4c, 1e
T80 60° turret				356c, 355e
90° turret				360e

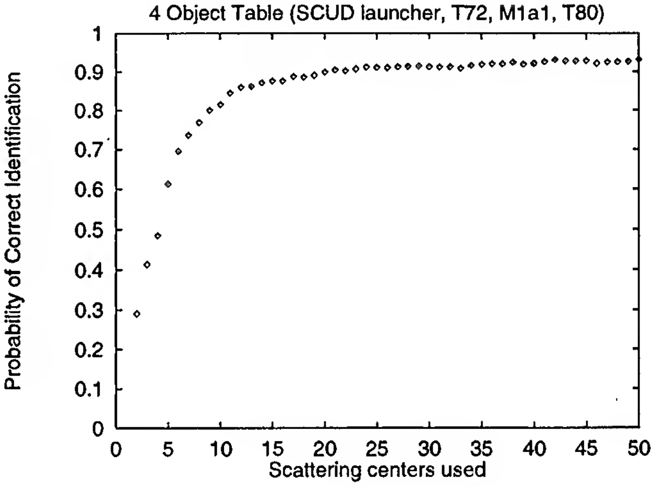


Figure 14: Effect of the number of scattering centers on articulated recognition rate.

articulations used in Tables 3 and 4, the plot of overall PCI vs number of scattering centers is shown in Figure 14 (each point is the result of 2520 trials). The maximum performance is achieved at 50 scattering centers (93.14%), but virtually the same performance could be found at 42 scattering centers (93.10%). A more optimal system with 35 scattering centers achieves similar performance, 92% PCI, with slightly less than half the storage and twice the speed of 50 scattering centers.

5.1.3 Articulation Invariance

The detailed recognition results can be related to the articulation invariance of articulated objects. The recognition failures for the T72 tank with the turret at 90° are plotted on the curve of percent invariance vs azimuth in Figure 15. These results show that recognition failures generally occur for azimuths where the percent invariance is low. Figure 16 shows how the PCI varies with the percent invariance. The points at low invariance values are misleading, because they

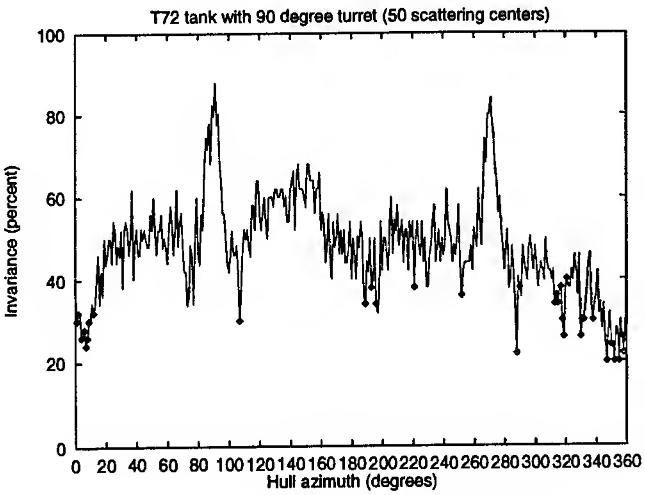


Figure 15: T72 tank (turret 90°) recognition failure plot on articulation invariance curve.

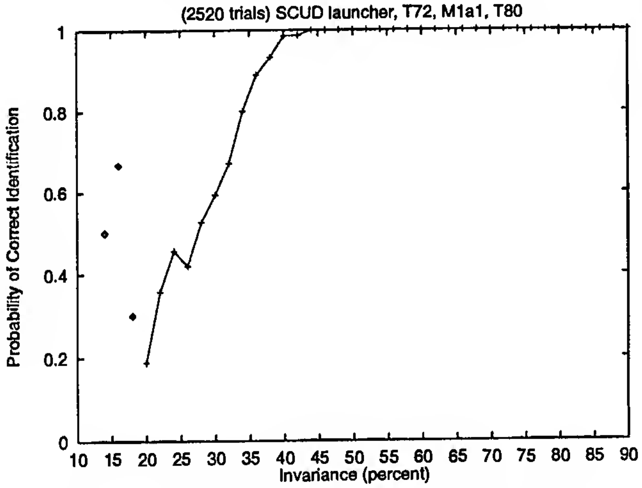


Figure 16: Recognition rate and articulation invariance (50 scatterers, average of 4 objects).

are due to a few correct identifications for the M1a1 tank, where the invariance (measured with respect to the hull) is low, yet a correct identification is made from features on the large turret. The overall recognition engine performance is almost perfect for invariance values greater than 40 percent (ie. down to 60 percent spurious data).

5.1.4 Identification (5 Object Table)

Table 5 shows results with a 5 object recognition table (50 scattering centers for each model), with the non-articulated FRED tank (which looks similar to the M1a1 tank, see Figure 17) added as a "confuser" in tests against the same 2520 articulated cases. In only four cases was a test object confused with the FRED tank: three times a T72 tank with 60° turret was now misidentified as a FRED tank, once a T72 tank with 90° turret, that had been misidentified as an M1a1 tank, was now misidentified as a FRED tank. The overall PCI for the 5 object table (with 50 scattering center models) was 93.02% versus 93.14% for the 4 object table.

Table 5: Confusion matrix for articulated identification results (5 object table, 50 scattering centers).

articulated	Non-articulated models				
test targets	SCUD	T72	M1a1	T80	FRED
	down	0° turret			
SCUD missile up	360				
T72 60° turret		332	7	18	3
90° turret		327	7	25	1
M1a1 60° turret		1	300	59	
90° turret		2	305	53	
T80 60° turret				360	
90° turret				360	

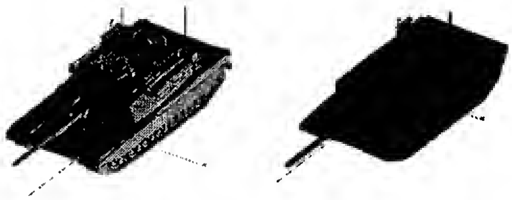


Figure 17: M1a1 (left) and FRED (right) tanks.

5.2 Occluded Objects

The occluded test data is simulated by starting with a given number of the strongest scattering centers and then removing the appropriate number of scattering centers encountered in order, starting in one of four perpendicular directions d_i (where d_1 and d_3 are the cross range directions, along and opposite the flight path respectively, and d_2 and d_4 are the up range and down range directions). Then the same number of scattering centers (with random magnitudes) are added back at random locations within the original bounding box of the chip. Each data set is 5760 test cases (4 objects X 4 directions X 360 azimuths). For the non-articulated occluded tests (the objects are the T72, M1a1, and T80 tanks with turret at 0° and the SCUD launcher with the missile down) there are 51 data sets (for 10, 30 and 50 scattering centers with 10 to 90% occlusion in 10% steps and the same for 20 and 40 scatterers plus 55, 65, and 75% occlusion) for a total of 293,760 test cases. Actually, only 50 data sets with a total of 288,000 test cases are used, because the data set of 10 scattering centers with 90% occlusion has less than two valid scattering centers for each case. For the articulated occluded tests the same tanks are used with a 90° turret and the missile is erect, but there are only 9 data sets (for 20 scattering centers with 10 to 90% occlusion) for a total of 51,840 test cases.

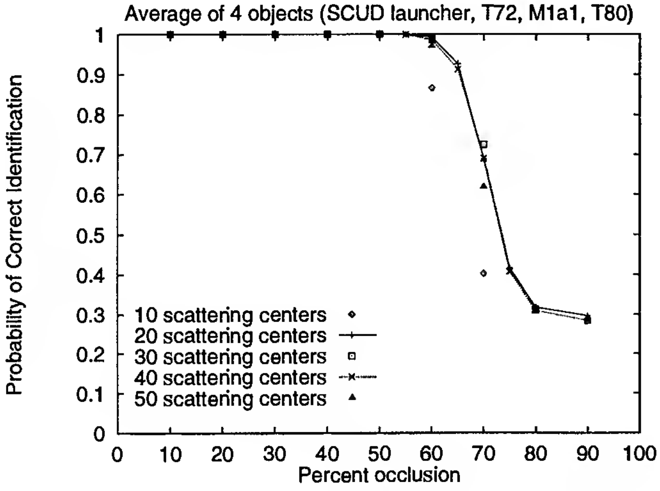


Figure 18: Recognition rate and occlusion percent.

5.2.1 Non-articulated Occluded Objects

The performance of the recognition engine with non-articulated occluded objects is shown in Figure 18 in terms of the probability of correct identification (PCI) as a function of percent occlusion with the 'number of scattering centers used' as a parameter. The results of 288,000 test cases are shown, where each point for a specific percent occlusion and number of scattering centers is the average PCI for all 4 occlusion directions, the 4 objects and the 360 azimuths (5760 test cases). The overall recognition engine performance is almost perfect for up to 60% occlusion. (This corresponds to results shown in Figure 16 for articulation invariance of 40% and above.) By 80 to 90% occlusion, the results are not much better than the 0.25 PCI one would expect by chance from the 4 possible objects. These performance results are replotted as Figure 19 to illustrate the effect of the number of scattering centers used on the recognition rate for the highly occluded cases. This indicates that optimal performance is in the region of 20 to 40 scattering centers.

5.2.2 Articulated Occluded Objects

Figure 20 shows the average and individual test object performance of the recognition engine (using 20 scattering centers) as a function of percent occlusion with 4 different articulated objects. The results of 51,840 test cases are shown. The overall performance for these articulated objects with 30% occlusion is a 0.698 PCI, almost the 0.70 system level goal [4] of the Moving and Stationary Target Acquisition and Recognition (MSTAR) program. The results are consistent with the average unoccluded articulated results for 20 scatterers, shown previously in Figure 14, which would be a 0.899 PCI at a "0%" occlusion in the occluded articulated results shown here in Figure 20. Figure 21 compares the performance results of the articulated and occluded articulated objects for cases with the same number of valid scatterers (i.e. 'scatterers used' in the unoccluded cases or 'unoccluded scatterers' in the occluded cases). In the occluded data the valid points are 'clustered' in a neighborhood which gets smaller as the occlusion increases (and the number of valid scatterers decreases). These relatively worse results for the naturally 'clustered' occluded articulated data, compared with the more widely distributed unoccluded articulated data (for the same number of valid scattering centers), illustrate the importance of the relatively rare long distances.

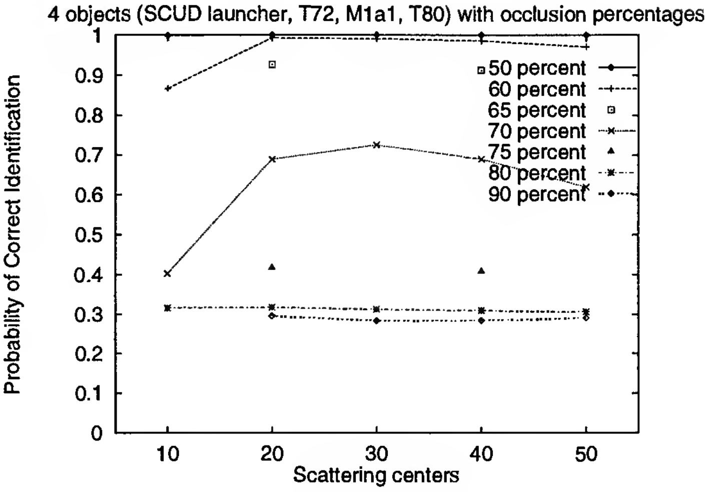


Figure 19: Effect of number of scatterers on occluded recognition rate.

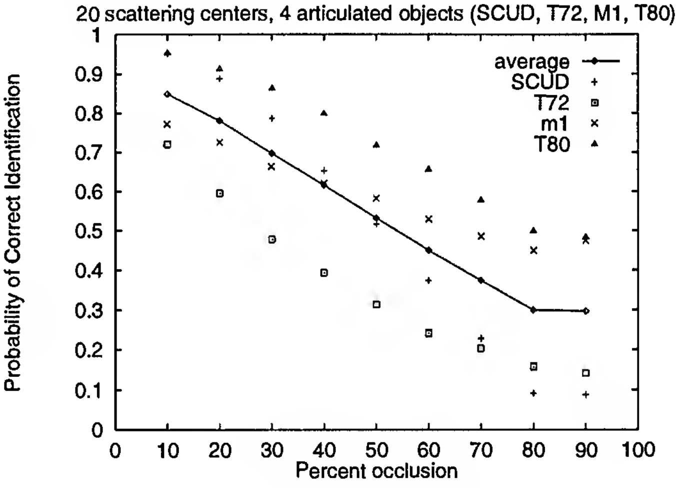


Figure 20: Articulated object recognition rate and occlusion percent.

6 Conclusions

The XPATCH generated, six inch resolution SAR imagery has great azimuthal variation that can be successfully captured by using 360 azimuth models for a given depression angle. Useful articulation invariant features are found in SAR images of military vehicles. The feasibility of a new concept for a SAR recognition engine to identify articulated and occluded objects based on non-articulated recognition models is

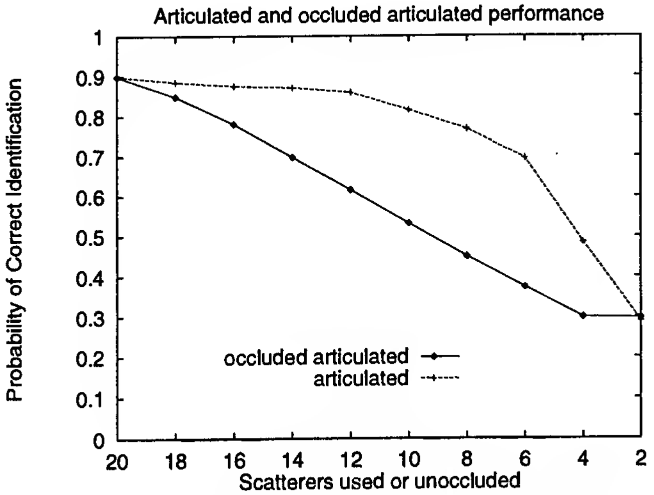


Figure 21: Articulated object and occluded articulated object performance results.

demonstrated. The performance of the recognition engine can be predicted by the percent articulation invariance (or percent unoccluded) when comparing the scattering center locations of the articulated (or occluded) test images with the non-articulated model scattering center locations. Limited experiments show that scaling to model more objects provides similar results, although performance will degrade depending on the number of coincidental similarities found in the radar signatures of the objects. Our results indicate the importance of the relatively rare long distances and suggest an explanation why this approach, which can use long distances (if available), could have an advantage over others [7] that are restricted to a "neighborhood".

Use of real SAR images of actual vehicles (vs XPATCH simulations from CAD models) would change the performance and detailed implementation of the design, but not the basic conceptual approach. Our approach of articulation invariance simply treats the articulated region as a "don't care", which applies to both real and simulated data. If real SAR images are more (or less) persistent in azimuth than XPATCH, then the recognition engine would need fewer (or more) azimuth models. Real SAR images and target chips are likely to have more noise than the ideal models and test data produced by XPATCH, however larger sets of noisy data can be used to produce useful recognition models. The noisy test data is manifest as some percentage of spurious data, which is similar to what was used to generate the occluded test data and the actual recognition results should suffer as indicated in Figures 18 and 20 on the Probability of Correct Identification vs percent occlusion curves (with the corresponding source of the invalid scattering centers being noise rather than 'occlusion').

Acknowledgment

The authors would like to thank Thomas Ferryman for his helpful suggestions.

References

[1] D. Andersch, S. Lee, H. Ling, and C. Yu. "XPATCH: A high frequency electromagnetic scattering prediction code using shooting and bouncing ray," *Proceedings of Ground Target Modeling and Validation Conference*, pp.498-507, Aug.1994.

[2] A. Beinglass and H. Wolfson. "Articulated object recognition, or: How to generalize the generalized Hough transform," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp.461-466, June 1991.

[3] B. Bhanu, G. Jones, J. Ahn, M. Li, and J. Yi. "Recognition of articulated objects in SAR images," *Proc. ARPA Image Understanding Workshop*, pp 1237-1250, Palm Springs CA, Feb 1996.

[4] T. Burns. "Moving and Stationary Target Acquisition and Recognition," *DARPA Image Understanding Technology Program Reviews*, Ft. Belvoir, VA, Sep.1996.

[5] D. Dudgeon, R. Lacoss, C. Lazott, and J. Verly. "Use of persistent scatterers for model-based recognition," *SPIE Proceeding: Synthetic Aperture Radar*, vol 2230, pp.356-368, Orlando, FL, April 1994.

[6] Y. Hel-Or and M. Werman. "Recognition and localization of articulated objects," *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 116-123, Austin, TX, Nov 11-12, 1994.

[7] K. Ikeuchi, T. Shakunga, M. Wheeler, and T. Yamazaki. "Invariant histograms and deformable template matching for SAR target recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 100-105, June 1996.

[8] Y. Lamden and H. Wolfson. "Geometric hashing: A general and efficient model-based recognition scheme," *Proc. International Conference on Computer Vision*, pp. 238-249, December 1988.

[9] S. Raney. "Automatic radar air to ground target acquisition program (Aragtap)." *ARPA BAA95-03: Moving and Stationary Target Acquisition and Recognition Algorithm Development, Briefing to Industry*, October 21, 1994.

[10] T. Strat. "Image understanding technology programs," *DARPA Image Understanding Technology Program Reviews*, Ft. Belvoir, VA, Sep.1996.

[11] J. Verly, R. Delanoy, and C. Lazott. "Principles and evaluation of an automatic target recognition system for synthetic aperture radar imagery based on the use of functional templates," *SPIE Proceedings: Automatic Target Recognition III*, vol 1960, pp. 57-71, Orlando, FL, April 1993.

Reinforcement Learning Integrated Image Segmentation and Object Recognition

Bir Bhanu, Xin Bao and Jing Peng
College of Engineering, University of California, Riverside, CA92521
Email: {bhanu, bao, jp}@constitution.ucr.edu
URL: http://constitution.ucr.edu/

Abstract

This paper presents a general approach to image segmentation and object recognition that can adapt the image segmentation algorithm parameters to the changing environmental conditions. Segmentation parameters are learned using a reinforcement learning (RL) algorithm that is based on a team of learning automata and operates separately in a global or local manner on an image. The edge-border coincidence is used as a short term reinforcement to reduce the computational expense due to model matching during the early stage of object recognition. However, since this measure is not reliable for object recognition, it is used later in conjunction with model matching in a closed-loop object recognition system that uses the results of model matching as a reinforcement signal in a "biased" learning system. The control switches between learning integrated global and local segmentation based on the quality of segmentation and model matching. Results are presented for both indoor and outdoor color images where the performance improvement is shown for both image segmentation and object recognition with experience.

1 Introduction

A model based object recognition system has three key components: image segmentation, feature extraction, and model matching. The goal of image segmentation is to extract meaningful objects from an input image. Image segmentation is an important and one of the most difficult low-level computer vision tasks [6]. All subsequent tasks including feature extraction, model matching, rely heavily on the quality of the image segmentation process.

The inability to adapt the image segmentation process to real-world changes is one of the fundamental weaknesses of typical model-based object recognition systems. Despite the large number of image segmentation algorithms available [10], no general methods have been found to process the wide diversity of images encountered in real world applications. Usually, an object recognition system is *open-loop*. Segmentation and feature extraction modules use default algorithm parameters, and generally work as pre-processing steps to the model matching component. The fixed sets of algorithm parameters used in various image segmentation and feature extraction algorithms generally degrade the system performance and lack adaptability in real-world applications. These default sets of algorithm parameters are usually obtained by the system designer by following a trial and error method. Parameters obtained in this way are not robust, since when the conditions for which they are designed are changed slightly, these algorithms generally fail without any graceful degradation in performance.

The usefulness of a set of algorithm parameters in a system can only be determined by the system's output, i.e., recognition performance. To recognize different objects or instances of the same object in an image, we may need different sets of parameters locally due to the changes in local image properties, such as brightness, contrast, etc. Also the changing environmental conditions (such as the time of the day, weather conditions, etc.), affect the appearance of an image which requires the capability to adapt the representation parameters for multi-scenario object recognition. To achieve robust performance in real-world applications, a need exists to apply learning techniques which can efficiently search image segmentation and feature extraction algorithm parameter spaces and find parameter values which yield optimal results for the given recognition task. In this paper, our goal is to develop a general approach to a learning integrated model-based object recognition system, which has the ability to continuously adapt to normal environmental variations.

In the remainder of the section 1, we present an overview of the approach, related work and the contributions of the paper. Section 2 gives the details of the approach and discusses algorithms used in this research. Section 3 provides the experimental results

*This work is supported by DARPA/AFOSR grant F49620-95-1-0424. The contents and information do not necessarily reflect the position or the policy of the U.S. Goverment.

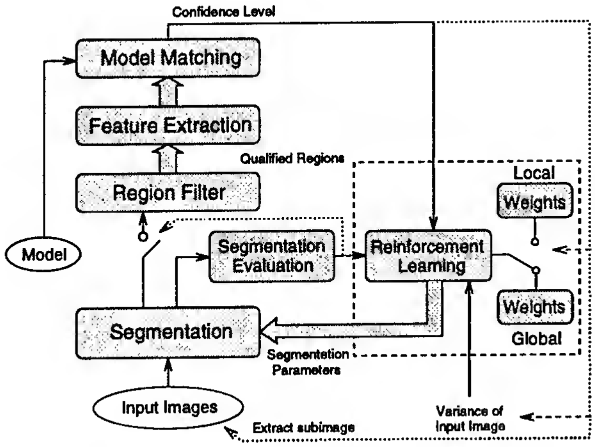


Figure 1: Reinforcement learning integrated image segmentation and object recognition system.

for segmentation and recognition on both indoor and outdoor color images. Finally, section 4 presents the conclusions and the future work.

1.1 Overview of the approach

In this paper, we present a general approach to reinforcement learning integrated image segmentation and object recognition. A reinforcement learning system is integrated into the model-based object recognition system to close the loop between model matching and image segmentation. The basic assumption is that we know the models of the objects that are to be recognized, but we do not know the number of objects and their locations in the image. The goal of the system is to maximize the matching confidence by finding a set of image segmentation algorithm parameters for the given recognition task (We have not discussed the problem of feature extraction parameters in this paper. It is described in a separate paper by Peng and Bhanu [1]). Thus, we address the problem of adaptive segmentation as finding a set of parameters for the given model and given input image. It reflects the fact that there may not exist a single set of "optimal" parameters which can be used for recognizing different objects in a given image. Figure 1 provides an overview of the system. Basically, the system consists of image segmentation, feature extraction, model matching, and reinforcement learning modules. The image segmentation component extracts meaningful objects from input images, feature extraction step performs polygonal approximation of connected components, and the model matching step tells us which regions in the segmented image contain the recognized object. The model matching module indirectly evaluates the performance of the image segmentation and feature extraction processes by generating a real valued matching confidence indicating the degree of success. This real valued matching confidence is then used to drive learning for image segmentation parameters in a reinforcement learning framework.

Given the computational expense for performing model matching, our approach uses edge-border coincidence [5] as a segmentation evaluation measure to find an initial point from which to begin the search through weight space. However, since this measure is not reliable as matching confidence, we use it in conjunction with model matching in a closed-loop system to adapt segmentation parameters to current input image conditions. Subsequent feature extraction and model matching are carried out for each connected component which passes through the size filter based on the expected size of objects of interest in the image. The highest matching confidence is taken as the reinforcement signal. Learning takes place as a result of interactions between segmentation and model matching.

Significant differences in characteristics exist between an image and its subimages, so operating conditions are tuned to these differences to achieve optimal performance of segmentation and model matching. For example, to recognize two objects in an image or a single object at different locations, it is often difficult, if not impossible, to meet all requirements with one process. It is essential to localize computation to meet each individual requirement. Thus, we adopt a control that switches between global and local segmentation phases based on the quality of image segmentation and model macthing.

The reinforcement learning integrated image segmentation and object recognition system is designed to be fundamental in nature and is not dependent on any specific image segmentation algorithms or type of input images. Reinforcement learning requires only the goodness of the performance rather than the details of algorithms that produce the results. To represent segmentation parameters suitably in a reinforcement learning framework, the system only needs to know the segmentation parameters and their ranges. In our approach, a binary encoding scheme is used to represent the segmentation parameters. While the same task could be learned in the original parameter space, for many types of problems, including image segmentation, the binary representation can be expected to learn much faster [2]. In this sense, the system is independent of a particular segmentation algorithm used.

1.2 Related work and our contributions

There is no published work on reinforcement learning integrated image segmentation and object recognition using multiple feedbacks. Bhanu and Lee [9] presented an image segmentation system which incorporates a genetic algorithm to adapt the segmentation process to changes in image characteristics caused by variable environmental conditions. In their approach, multiple segmentation quality measures are used as feedback. Some of these measures require ground-truth information which may not be always available. Peng and Bhanu [2] presented an approach in which a reinforcement learning system is used to close the loop between segmentation and recognition, and to induce a mapping from input images to corresponding segmentation parameters. Their approach is based on global image segementation which is not the best way to detect objects in an image; we need the capability of performing segmentation based on local image properties (local segmentation). Another disadvantage of their method is its time complexity which makes it problematic for practical application of computer vision.

For object recognition applications, the efficiency of the learning techniques is very important. How to add bias, a prior or domain knowledge in a reinforcement learning based system is an important topic of research in reinforcement learning [3][7] [8]. For the *RATLE* system, Maclin and Shavlik [3] accept "advice" expressed in a simple programming language. This advice is compiled into "knowledge-based" connectionist Q-learning network. They show that advice-giving can speed up Q-learning when the advice is helpful (though it need not be perfectly correct). When the advice is harmful, back propagation training quickly overrides it. Dorigo and Colombetti [7] show that by using a learning technique called learning classifier system (LCS), an external trainer working within a RL framework can help a robot to achieve a goal. Thrun and Schwartz [8] have discussed methods for incorporating background knowledge into a reinforcement learning system for robot learning.

In our approach, the edge-border coincidence is used to locate an initial good point from which to begin the search through weight space for high matching confidence values. Although as a segmentation evaluation measure the edge-border coincidence is not as reliable as the matching confidence, lower edge-border coincidence values always result in poor model matching. Likewise, higher edge-border coincidence values suggest with high probability that the current set of segmentation parameters is in a close neighborhood of the optimal one. It is an inexpensive way to arrive at an initial approximation to a set of segmentation parameters that gives rise to the optimal recognition performance. The control switches between global and local segmentation processes to optimize recognition performance. To further speed-up the learning process the reinforcement learning is biased when the model matching confidence or the edge-border coincidence is used as the reinforcement signal (note that the reinforcement learning is unbiased *initially* when the edge-border coincidence is used as the reinforcement signal). We achieve better computational efficiency of the learning system and improved recognition rates compared to the system developed by Peng and Bhanu [2].

The original contributions of the reinforcement learning integrated image segmentation and object recognition system presented in this paper are:

- To achieve *robustness* for image recognition system operating in real world, model matching confidence is used as feedback to influence the image segmentation process, and thus provide an adaptive capability.
- A RL system based on a team of learning automata is applied to represent and update *both* global and local image segmentation parameters. The learning system optimizes segmentation performance on each individual image and accumulates segmentation experience over time to reduce the effort needed to optimize future unseen images.
- Edge-border coincidence, as a segmentation evaluation measure, reduces computational costs by avoiding expensive model matching, especially during earlier stages of learning.
- Learning local segmentation parameters on subimages, which may potentially contain objects, improves the performance of object recognition system.
- Explicit bias is used in the RL based system to speed up the learning process for adaptive image segmentation.

2 Technical Approach

The goal of our system is to maximize the model matching confidence by finding a set of image segmentation algorithm parameters for a given recognition task. To reduce the computational expense of model matching, the edge-border coincidence is first used as evaluation function to find a set of parameters from which to begin the learning. The segmentation process has two distinct phases: global and local. While global segmentation is performed for the entire image, local segmentation is carried out only for selected subimages. For a set of input images, the system takes inputs sequentially. This is similar to human visual learning process, in which the visual stimulus are presented temporally in a sequential manner. For the first input image, since the system has no accumulated experience, we initialize the system using random value of weights in the unbiased stocastic RL algorithm. For each input image thereafter, the learning process starts from the set of segmentation parameters learned based on all the previous input images. The following are the main steps of our learning algorithm:

Initial Approximation. The edge-border coincidence is used as a short term reinforcement during earlier stages of learning to drive weight changes without going through the expensive model matching process. Once the edge-border coincidence has exceeded a given threshold, the weight changes will be driven by the matching confidence, which requires more expensive computation of feature extraction and model matching.

Learning Global Segmentation. A network of biased Bernoulli units generates a set of segmentation parameters from which segmentation is performed on the entire image. The evaluation of the segmentation process is provided by the model matching confidence, which is then used to drive changes to the weights according to the reinforcement learning algorithm. We assume that we have a prior knowledge of the size of objects of interest in the images. For those connected components which pass through the size filter based on the expected size of objects of interest in the image, we perform feature extraction and model matching. The highest matching confidence is taken as the reinforcement to the learning system. If the highest matching confidence level is above a given switching theshold, we focus image segmentation and model matching on the connected component and switch to the local search process.

Learning Local Segmentation. Once a connected component has been extracted from the input image, the local search begins to find the best fit parameters for the subimage. It starts from the current estimate of weights that resulted from global learning. Similar to global learning, the matching confidence is used to update the weights estimate, until the matching confidence reaches the accepting threshold (0.8 in our experiments) or the number of iterations reaches the *MaxLocal* (in our experiments, it is set at 20). If after *MaxLocal* loops, the matching confidence is still under the accepting threshold, we switch back to the global learning process, continue the learning from where we switched to the local search process. If the matching confidence reaches the accepting threshold, the learning process for the current input image is terminated.

2.1 *Phoenix* image segmentation algorithm

Since we are working with color imagery in our experiments, we have selected the *Phoenix* segmentation algorithm [16] [18] developed at Carnegie-Mellon University and SRI International. The *Phoenix* segmentation algorithm has been widely used and tested. It works by recursively splitting regions using histogram for color features. *Phoenix* contains seventeen different control parameters, fourteen of which are adjustable. The four most critical ones that affect the overall results of the segmentation process are selected for adaptation: *Hsmooth, Maxmin, Splitmin*, and *Height*. *Hsmooth* is the width of the histogram smoothing window. *Maxmin* is the lowest acceptable peak-to-valley height ratio. *Splitmin* represents the minimum area for a region to be automatically considered for splitting. *Height* is the minimum acceptable peak height as a percentage of the second highest peak. Each parameter has 32 possible values. The resulting search space is 2^{20} sample points. Each of the *Phoenix* parameters is represented using 5 bit binary code, with each bit represented by one Bernoulli unit. To represent 4 parameters, we need a total of 20 Bernoulli units. More details about *Phoenix* are given in the report by Laws [16].

Table 1: Ranges for selected *Phoenix* parameters.

Parameter	Sampling Formula	Range
Hsmooth: hs $\in$ [0 : 31]	hsmooth=1 + 2 × hs	1 – 63

Maxmin:
mm $\in$ [0 : 31]
	ep=ln(100)+0.05 × mm
	maxmin = exp(ep) + 0.5
		100 – 471

Splitmin: sm $\in$ [0 : 31]	splitmin=9 + 2 × sm	9 – 71
Height: h $\in$ [0 : 31]	height=1 + 2 * h	1 – 63

2.2 Segmentation evaluation

Given that feature extraction and model matching are computationally expensive processes, it is imperative that initial approximation be made such that overall computation can be reduced. To achieve this objective, we introduce a secondary feedback signal - segmentation evaluation that evaluates the image segmentation quality. There are a large number of segmentation quality measures that have been suggested. The segmentation evaluation we selected is the *edge-border coincidence* [9][17], which measures the overlap of the region borders in the segmented image relative to the edges found using an edge detector, and does not depend on any ground-truth information. In this approach, we use the *Sobel* edge detector to compute the necessary edge information. Edge-border coincidence is defined as follows. Let E be the set of pixels extracted by the edge operator and S be the set of pixels found on the region boundaries obtained from the segmentation algorithm:

$$\text{Edge} - \text{border coincidence} = \frac{n(E \cap S)}{n(E)},$$

where $n(A)$ is the number of elements in set A

Figure 2 shows the Sobel edge image of an experimental indoor color image and the boundaries of the segmented image using the *Phoenix* segmentation algorithm. The edge-border coincidence for the segmented image is 0.6825. Segmentation evaluation indicates the quality of the segmentation process. Matching confidence, the recognition system's output, indicates the confidence of the model matching process, and indirectly shows the segmentation quality of the recognized object. It is possible that

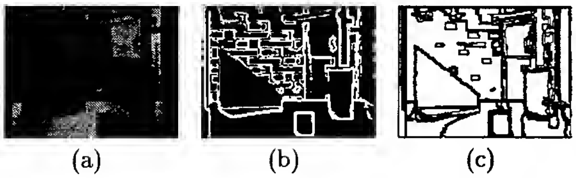


Figure 2: Edge-border coincidence. (a) input image; (b) Sobel edge magnitude image (threshold = 200); (c) boundaries of the segmented image. Segmentation parameters are: *Hsmooth*=7, *Maxmin*=128, *Splitmin*=47, *Height*=60.

segmentation evaluation is high and matching confidence level is low, or segmentation evaluation is low and matching confidence is high. Figure 3(a) shows that global segmentation evaluation is not well correlated with matching confidence. However, local segmentation evaluation, which measures the overlap between the edges and region borders of a subimage, is strongly correlated to the matching confidence, as shown in Figure 3(b).

Although the global segmentation evaluation does not correctly predict the matching confidence, for our purpose it is sufficient to drive initial estimates. If the edge-border coincidence is under a threshold, which indicates a low possibility to get a good recognition result, the system repeats the initial estimation process using the edge-border coincidence as the sole reinforcement feedback signal until the edge-border coincidence is greater than the threshold. At that time, the segmentation performance will be determined completely by the model matching.

2.3 Reinforcement learning for image segmentation

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. It is appropriately thought of as a class of problems, rather than as a set of techniques [4]. This type of learning has a wide variety of applications, ranging from modeling behavior learning in experimental psychology to building active vision systems. The term *reinforcement* comes from studies of animal learning in experimental psychology. The basic idea is that if an action is followed by a satisfactory state of affairs or an improvement in the state of affairs, then the tendency to produce that action is reinforced. Reinforcement learning is similar to supervised learning in that it receives a feedback to adjust itself. However, the feedback is *evaluative* in the case of reinforcement learning. In general, reinforcement learning is more widely applicable than supervised learning and it provides a competitive approach to building autonomous learning systems that must operate in real world.

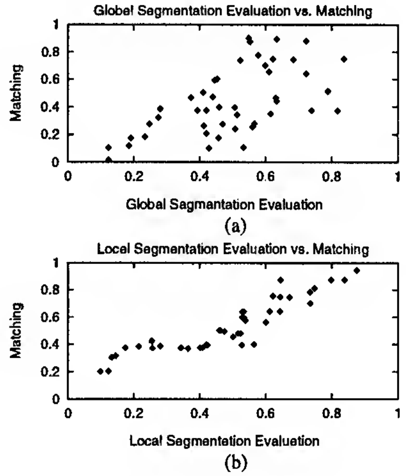


Figure 3: (a) Global edge-border coincidence vs. matching confidence; (b) Local edge-border coincidence vs. matching confidence for recognizing the cup in the image shown in Figure 2(a).

There are several reasons why we apply reinforcement learning in our computer vision system. *First*, reinforcement learning requires knowing only the goodness of the system performance rather than the details of algorithms that produce the results. In the object recognition system, model matching confidence indirectly evaluates the performance of image segmentation and feature extraction processes. It is a natural choice to select matching confidence as a reinforcement signal. *Second*, convergence is guaranteed for several reinforcement learning algorithms. *Third*, reinforcement learning is well suited to the multi-level object recognition problems in image understanding. It can systemically assign rewards to different levels in a computer vision system.

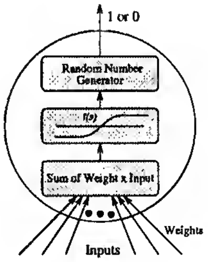


Figure 4: Basic structure of a Bernoulli unit.

The particular class of reinforcement learning algorithms employed in our system is the connectionist *REINFORCE* algorithm [11], where units in such a network are *Bernoulli quasi-linear units*. Figure 4 shows the basic structure of a Bernoulli unit. A team of five independent Bernoulli units represent a segmentation parameter with 32 possible values. The output of each unit is either 1 or 0, determined stochastically using the Bernoulli distribution with probability mass function $p = f(s)$, where f is the logistic function. For such an unit, p represents the probability of choosing 1 as its output value.

$$f(s) = \frac{1}{1+e^{-s}}, \quad where\ s = \sum_j w_{ij}x_j$$

where w_{ij} is the weight of the jth input for unit i, and x_j is the jth input value for each unit. In the reinforcement learning paradigm, the learning component uses the reinforcement $r(t)$ to drive the weight changes according to a particular reinforcement learning algorithm used by the network. The specific algorithm we used has the following form: for each unit, at the tth time step, after generating output $y(t)$ and receiving reinforcement signal $r(t)$, increment each weight w_{ij} by

$$\Delta w_{ij}(t) = \alpha[r(t) - \bar{r}(t-1)][y_i(t) - \bar{y}_i(t-1)]x_j - \delta w_{ij}(t)$$

where α is the learning rate, δ is the weight decay rate, x_j is the input to each Bernoulli unit, y_i is the output of the ith Bernoulli unit. The term $r(t) - \bar{r}(t-1)$ is called the *reinforcement factor*, and $y_i(t) - \bar{y}_i(t-1)$ is the *eligibility* of the weight w_{ij}. $\bar{r}(t)$ is the exponentially weighted average of prior reinforcement values,

$$\bar{r}(t) = \gamma\bar{r}(t-1) + (1-\gamma)r(t), \quad with\ \bar{r}(0) = 0$$

γ is the trace parameter. Similarly, $\bar{y}_i(t)$ is an average of past values of y_i computed by the same exponential weighted scheme used for $\bar{r}(t)$,

$$\bar{y}_i(t) = \gamma\bar{y}_i(t-1) + (1-\gamma)y_i(t)$$

The algorithm has the convergence property [11] such that it statistically climbs the gradient of expected reinforcement in weight space. The weight decay is used as a simple method to force the sustained exploration of the weight space.

Note that a team of 20 Bernoulli units represents the four image segmentation parameters selected for learning. Each bit of a parameter is independent of each other. Thus, it allows us to search the parameter space thoroughly.

2.4 Feature extraction and model matching

Feature extraction consists of finding polygon approximation tokens for each connected component obtained after image segmentation. To speed up the learning process, we assume that we have the prior knowledge of the *approximate size* (area) of the object, and only those connected components whose area (number of pixels) are comparable with the area of the model object are approximated by a polygon. In Figure 1, the *region filter* selects those connected components whose areas are in the expected range. For example, in our experiment on indoor images, the cup is the target object. The expected area is from 200 to 450 pixels. Figure 5 shows the boundaries of a segmented image, selected regions whose areas are in the expected range, and the polygon approximation of these regions. The polygon approximation is implemented by calling the polygon approximation routine in *Khoros* [12]. The resulting polygon approximation is a vector image to store the result of the linear approximation. The image contains two points for each estimated line. The polygon approximation has a fixed set of parameters:

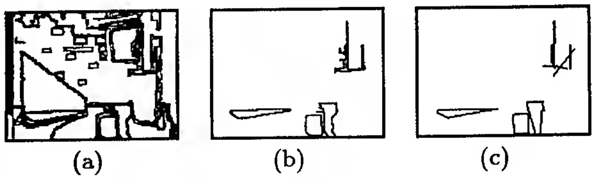


Figure 5: (a) Boundaries of the segmented image shown in Figure 2(a)(segmentation parameters are: *Hsmooth*=7, *Maxmin*=128, *Splitmin* =47, *Height*=54). (b) Selected regions whose areas are in the expected range (200 – 450 pixels), (c) Polygon approximation of these regions (parameters as specified in this section).

- Minimal segment length for straight line - 5. When the estimated straight line has a length less than this threshold, it is skipped over.
- Elimination percentage - 0.1. Percentage of line length rejected to calculate parameters of the straight line.
- Approximation error - 0.6. Threshold Value for the approximation error. When the calculated error is greater than this value, the line is broken.

Model matching employs a cluster-structure matching algorithm [14] which is based on forming the clusters of translational and rotational transformations between the object and the model. The algorithm takes as input two sets of tokens, one of which represents the stored model and the other represents the input region to be recognized. It then performs topological matching between the two token sets and computes a real number that indicates the confidence level of the matching process. Basically, the technique consists of three steps: clustering of border segment transformations; finding continuous sequences of segments in appropriately chosen clusters; and clustering of sequence average transformation values. More details about this algorithm are given in [14].

2.5 Biased reinforcement learning for image segmentation

In the RL algorithm as described in section 2.3, each of the bits of each of the parameters is independent. The output of each bit depends on the value of p, which represents the probability of an unit to choose

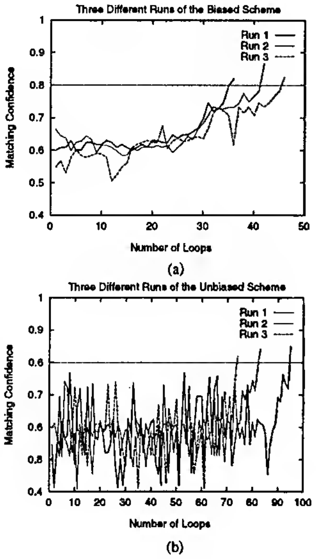


Figure 6: Matching confidence history of three runs of the biased and unbiased RL algorithms on the image shown in Figure 2(a). (a) biased; (b) unbiased.

1 as its output. In the initialization phase, we use the *unbiased* RL algorithm in which the output of each bit of a parameter is determined in the following way:

$$y_i = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$$

It is "unbiased" in that the output of a bit is governed solely by the Bernoulli probability law. The advantage is that rapid changes in output values allow giant leaps in the search space, which in turn enables the learning system to quickly discover suspected high pay-off regions. However, once the system has arrived at the vicinity of a local optimum, as will be the case after the initial estimation, changes in the most significant bit will drastically alter the parameter value, often jumping out of the neighborhood of the local optimum. Ideally, once the learning system discovers that it is within a possible high pay-off region, it should attempt to capture the regularities of the region. This then biases future search toward points within it. The challenge, of course, is to have a learning algorithm that allows the parameters controlling the search distribution to be adjusted so that this distribution comes to capture this knowledge. The algorithm described here shows some promise in this regard. In order to force parameters to change slowly, after the initialization phase, we apply a *biased* RL algorithm in which the two most significant bits of a parameter are forced to change in a slower fashion as:

$$y_i = \begin{cases} 1 & \text{if } p > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

and other bits use the the same rule as described in the unbiased RL algorithm. Figure 6 shows the experimental results of the two schemes on the image shown in Figure 2(a). In this experiment, we only apply the initialization followed by global learning without switching between global and local learning. The results show that the biased RL algorithm demonstrates a speed up of 2 – 3.

```
procedure Initialization( )
    generate a set of random weights
    repeat
        compute the segmentation parameters
        segment the image and compute edge-border coincidence
        r(i) = edge-border coincidence, update weights
    until edge-border coincidence > EB1

procedure Global_Segmentation( )
    r(0) = 0.5,  highest_matching_confidence = 0
    for i from 1 toMaxGlobal do
        for each connected component which passes the size filter do
            feature extraction and model matching
            if matching_confidence > Switch then
                Local_Segmentation( )
            if matching_confidence > highest_matching_confidence
                highest_matching_confidence = matching_confidence
        r(i) = highest_matching_confidence
        if recognized all the connected components then exit
        count = 0
        repeat
            compute the segmentation parameters using r(i)
            segment the image using the current set of parameters
            count++
            if count > MaxSeg then exit
        until edge-border coincidence > EB2

procedure Local_Segmentation( )
    extract subimage from the input image
    compute standard deviations of parts of the subimage
    copy the weights from global to local process
    count = 0
    while count < MaxLocal do
        subimage segmentation, feature extraction, and model matching
        update weights using matching confidence as reinforcement
        if matching confidence > Accept then recognized and return
        count++
```

Figure 7: Algorithm description.

2.6 Algorithm description

Figure 7 shows the implementation of our algorithm. The algorithm works by switching between global and local segmentation. Initially, if the system has no accumulated knowledge, the edge-border coincidence is used as the evaluation function to search a set of image segmentation parameters using unbiased reinforcement learning algorithm. Otherwise, the input image is segmented using the set of parameters learned from previous images. *EB1* and *EB2* are two thresholds for edge-border coincidence. During the initial unbiased reinforcement learning phase, if the edge-border coincidence is greater than *EB1* (= 0.5 in our experiments), then we can start

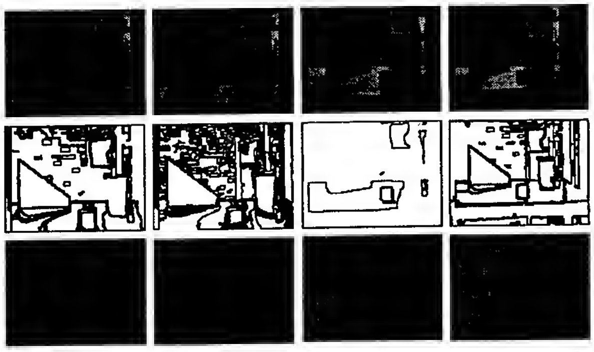


Figure 8: Row 1: input images; row 2, 3: corresponding segmented image and recognized object. For each input image, global segmentation evaluation, local segmentation evaluation for the selected object, and matching confidence are (0.67, 0.74, 0.87); (0.87, 0.62, 0.93); (0.22, 0.82, 0.91); (0.68, 0.73, 0.92). The learned *Phoenix* segmentation parameters *Hsmooth, Maxmin, Splitmin*, and *Height* after local learning process are (7 122 47 52); (7 128 47 52); (5 471 19 58); (11 192 59 48).

the learning process with a high expectation to generate good recognition results. During the global segmentation phase, if the segmentation quality is less than *EB2* (= 0.4 in our experiments), the object is less likely to be present in the segmented image, and choosing another set of parameters using the biased RL algorithm with the current reinforcement signal can speed up the process.

In the global segmentation procedure, if the global segmentation loops more than *MaxGlobal*, we conclude that the object does not appear in the image and terminate the learning process for the given input image. For each connected component which passes the region filter, if the matching confidence is greater than *Switch*, then we can switch the control from global to local segmentation. During local segmentation, if the matching confidence reaches *Accept*, we conclude that the connected component is the recognized model object. If the local segmentation loops more than *MaxLocal*, the control will switch back to global segmentation since the object is not likely to be extracted in the subimage and we resume the global segmentation process.

3 Experimental Results

The system is verified through a set of 12 indoor and a set of 12 outdoor color images. These images are acquired at different times and different viewing distances with varying lighting conditions. The size of indoor images is 120 by 160 pixels, and the size of outdoor images is 120 by 120 pixels. Each image is decomposed into 4 images for *Phoenix* segmentation – red, green, blue components, and the Y component

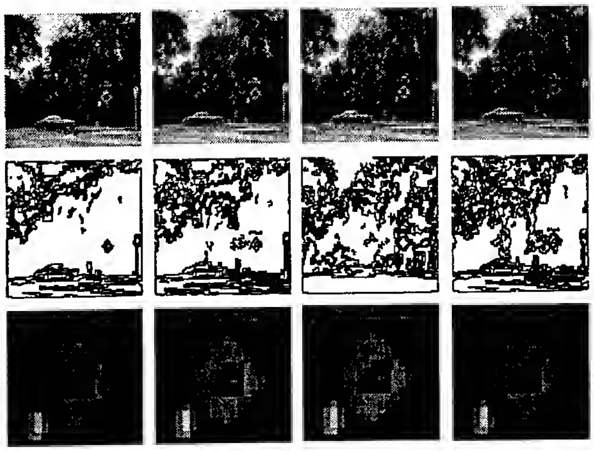


Figure 9: Row 1: input images; row 2, 3: corresponding segmented image and recognized object. For each input image, global segmentation evaluation, local segmentation evaluation for the selected object, and matching confidence are (0.59, 0.51, 0.82); (0.79, 0.57, 0.85); (0.85, 0.76, 0.88); (0.82, 0.53, 0.92). The learned *Phoenix* segmentation parameters *Hsmooth, Maxmin, Splitmin*, and *Height* after local learning process are (11 367 43 26); (11 259 23 46); (11 259 29 56); (9 276 31 46).

of YIQ model of color images. For the indoor images, the desired object is the cup in the image, and in the outdoor images, the target object is the traffic sign. The expected size of the cup and the traffic sign are 200 to 450 pixels and 36 to 100 pixels, respectively.

Based on the size of the object to be recognized in the image, we divide the Y component image into 48 subimages for the indoor images, and 36 subimages for the outdoor images. Each subimage's size is 20 by 20 pixels. The standard deviations of those subimages serve as inputs to each Bernoulli unit, i.e., each Bernoulli unit has a total of 48 inputs (and therefore, 48 weights) for the indoor image, and has a total of 36 inputs (36 weights) for the outdoor image. To learn the four selected *Phoenix* segmentation parameters, we need 20 Bernoulli units. So there is a total of 960 weights for indoor images, and 720 weights for outdoor images.

For the team of 20 Bernoulli units, the parameters α, γ, and δ are determined empirically, and they are kept constant for all images. In our experiments, $\alpha = 0.02$, $\gamma = 0.9$, and $\delta = 0.01$, *EB1* = 0.5, *EB2* = 0.4, *MaxGlobal, MaxLocal*, and *MaxSeg* are all set to 20. The threshold for matching confidence *Switch* = 0.6, and *Accept* = 0.8. Threshold used for extracting edges using Sobel operator is set at 200.

3.1 Results on indoor and outdoor images

Figure 8 and 9 show the experimental results on the

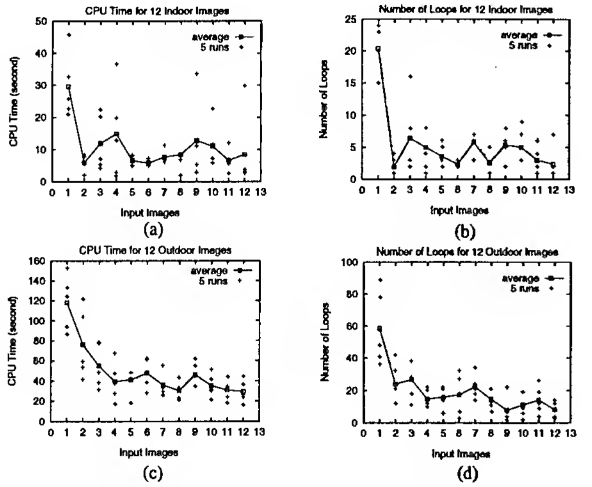


Figure 10: (a) CPU time for 5 different runs on 12 indoor images and the average; (b) Number of loops for 5 different runs on 12 indoor images and the average; (c) CPU time for 5 different runs on 12 outdoor images; (d) Number of loops for 5 different runs on 12 outdoor images.

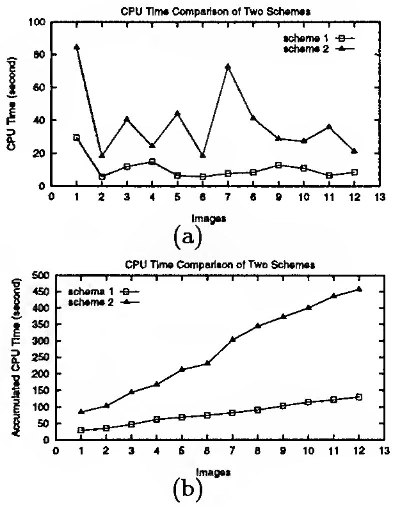


Figure 11: Comparison of two approaches: scheme 1–approach presented in this paper, scheme 2–Peng and Bhanu's approach [2]. (a) Comparison of the average CPU time of 5 different runs on 12 indoor images; (b) Comparison of the acumulated average CPU time of 5 different runs on 12 indoor images.

set of 12 indoor color images and the set of 12 outdoor color images. For each indoor image, the globally segmented image using the set of learned parameters and the extracted object which has been finally recognized, are presented. For each set of images, the 12 images are taken sequentially. Except for the first image, the learning process for each image starts from the global segmentation parameters learned from all the previous images. For the first input image, the learning system is initialized using the unbiased RL algorithm. Usually, it takes less than 45 iterations to find a set of segmentation algorithm parameters which produces high edge-border coincidence. Figure 8 and 9 also show the global edge-border coincidence, local edge-border coincidence, model matching confidence, and the four learned segmentation parameters after local learning process for each input image.

Figure 10 shows the CPU time for the 12 indoor images and 12 outdoor images for five different runs, and the number of loops for each input image, which is the sum of all the loops involved in the global learning and local learning processes. These two curves show the learning capability of the system, i.e., the system uses less and less CPU time with experience to find a set of segmentation parameters and correctly recognizes the object. The number of learning loops decreases with the accumulation of experience.

3.2 Comparison of the two approaches

In this section we compare the performance of our system as shown in Figure 1 with the approach discussed in the paper by Peng and Bhanu [2]. We show the effect of incorporating segmentation evaluation using the edge-border coincidence into the learning system and the impact of global and local segmentations on model matching.

The key differences between the two methods are the introduction of the local segmentation process, the biasing of RL algorithm, and the use of edge-border coincidence as an evaluation of the segmentation performance during earlier stages of learning in order to reduce the computational expense stemming from model matching. The segmentation process alternates between the whole image and its subcomponents. The local segmentation is highly desirable when there are multiple targets or a single target at multiple locations with different local characteristics. It can dramatically improve the recognition performance. The biasing of RL algorithm reduces computational time as illustrated in Figure 6.

In the paper by Peng and Bhanu [2], the matching confidence is the only feedback that drives learning. Although it is undoubtedly the most reliable measure, it is relatively expensive to compute. Here the edge-border coincidence provides us with a cheap way to find a good point from which to begin the more expensive search for high matching confidence values. Figure 11 shows the comparison results of the two schemes: our scheme (scheme 1) and Peng and Bhanu's scheme (scheme 2). Although good initial estimates may not always result in faster discovery of high matching confidence values, the edge-border coincidence seems to work well in practice for all the problems we have experimented.

4 Conclusions and future work

We have presented a proof-of-the-principle of a general approach for adaptive image segmentation and object recognition. The approach combines a domain independent simple measure for segmentation evaluation (edge-border coincidence) and domain dependent model matching confidence in a reinforcement learning framework in a systematic manner to accomplish robust image segmentation and object recognition simultaneously. Experimental results demonstrate that the approach is suitable for continuously adapting to normal changes encountered in real-world applications.

For adapting to the wide varity of images encountered in real-world applications, we can develop an autonomous gain control system which will allow the matching between different classes of images taken under significantly different weather conditions (sun, cloud, snow, rain) and adapt the parameters within each class of images. We use image context to divide the input images into several classes based on image properties and external conditions, such as time of the day, lighting condition, etc. [9]. When an image is presented, we use an image property measurement module and the available external information to find the stored information for this category of images, and start learning process from that set of parameters. This will overcome the problem of adapting to large variations between consecutive images.

The real significance of using a learning network to select segmentation parameters to optimize model matching performance is that interconnections within the network can enforce coordination of the choices made by the output units in order to concentrate the search in suspected high-payoff regions of the parameter space. A network that can coordinate the choices made by the output units should be able to generate certain combinations of bits with greater probability than if their individual components were selected independently. If the network operates in this way it should expect to find high matching confidence values much more quickly than without coordination. We plan to explore these issues in the future.

References

[1] J. Peng and B. Bhanu. Delayed reinforcement learning for closed-loop object recognition. In *Proc. of the 13th Inter. Conf. on Pattern Recognition*, pp. 310-314, Vienna, Austria, Aug. 1996.

[2] J. Peng and B. Bhanu. Closed-loop object recognition using reinforcement learning. In *Proc. IEEE Computer Society Confernece on Computer Vision and Pattern Recognition*, pp. 538-543, San Francisco, CA, June 1996.

[3] R. Maclin, J. W. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251-281, 1996.

[4] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intell. Research*, 4:237-285, 1996.

[5] B. Bhanu, S. Lee and J. Ming. Adaptive image segmentation using a genetic algorithm. *IEEE Trans. on Systems, Man and Cybernetics*, 25(12):1543-1567, Dec. 1995.

[6] B. Bhanu, S. Lee and S. Das. Adaptive image segmentation using genetic and hybrid search methods. *IEEE Trans. on Aerospace and Electronic Systems*, 31(4):1268-1291, Oct. 1995.

[7] M. Dorigo and M. Colombetti. Robot shaping: Developing autonomous agents through learning. *Artificial Intelligence*, 71(2):321-370, Dec. 1994.

[8] S. Thrun and A. Schwartz. Finding structure in reinforcement learning. In *Proc. of Advances in Neural Information Processing Systems 7*, pp. 385-392, Denver, CO, 1994.

[9] B. Bhanu and S. Lee. *Genetic Learning for Adaptive Image Segmentation.* Boston MA: Kluwer Academic Publishers, 1994.

[10] B. Bhanu and T. L. Jones. Image understanding research for automatic target recognition. *IEEE Aerospace and Electronics Systems Magazine*, 8(10):15-23, Oct. 1993.

[11] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229-256, 1992.

[12] J. Rasure and D. Argiro. Khoros User's Manual. University of New Mexico. 1991

[13] A. G. Barto, R. S. Sutton and C. J. C. H. Watkins. Learning and sequential decision making. COINS Technical Report 89-95. Department of Computer and Information Science, University of Massachusetts, Amherst, MA, 1989.

[14] B. Bhanu and J. Ming, Recognition of occluded objects: A cluster-structure algorithm. *Pattern Recognition*, 20(2):199-211, 1987.

[15] S. Shafer and T. Kanade. Recursive region segmentation by analysis of histograms. In *Proc. of IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1166-1171, 1982.

[16] K. Laws. The *Phoenix* image segmentation system: Description and evaluation. SRI International Technical Report TR289, Dec. 1982.

[17] D. L. Milgram. Region extraction using convergent series. *Computer Graphics and Image Processing*, 11:1-12, 1979.

[18] R. Ohlander, K. Price and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313-333, 1978.

Target Detection in UWB SAR Images Using Temporal Fusion

Li-Kang Yen, José C. Principe
Computational NeuroEngineering Laboratory
Electrical and Computer Engineering Department
University of Florida
http://www.cnel.ufl.edu
yen@cnel.ufl.edu, principe@cnel.ufl.edu

Abstract

The new ultra wide band (UWB) synthetic aperture radar (SAR) exhibits a different reflection phenomenology on metallic surfaces which is characterized by a resonant response. In order to capture the information contained in the resonant response we propose to project the down range profiles in a Laguerre function space, which is an orthogonal expansion of decaying exponentials. However, the large energy contained in the driven response (from both metallic and non-metallic objects) would degrade the performance of CFAR detectors based on spatial information. We propose to sequentially combine two subspace CFAR detectors, one to detect the driven response and the other to detect the resonance response. A neural network is utilized to fuse the two responses. Preliminary results corroborate the adequacy of the method.

1.0 Introduction

It is well known that man-made metallic objects in UWB radar produce a delayed damped sinusoidal response following the large energy reflection called the driven response. The driven response is also generated by non-metallic objects such as trees or foliage. CFAR algorithms based on generalized likelihood ratio test (GLRT) have been proposed for detecting targets resonance response in UWB SAR images [Yen and Principe, 1997a][Yen and Principe, 1997b], but they utilize only either the resonance response information or a combination of the resonance response and the spatial extent of the target. However, the large energy contained in the driven response of non-metallic objects like foliage degrades the performance of these CFAR detectors.

In this paper we propose the strategy of temporally combining two CFAR detectors [Wang and Chellapa 1994], one for detecting the early driven response and the other for detecting the delayed resonance response. The two detections are further fused by a neural network. Although some algorithms were proposed to "optimally" fuse several statistically independent detections [Chair and Varshney, 1996], they are either only optimal for fixed threshold local detectors or for unrealistic assumptions about data distributions. In this paper, the previous fusion rule in are extended using the neural network, so that all the weights and threshold could be adapted by the training data to perform better.

In section II, we discuss the temporal template model for target's driven response and resonance response. In section III, we formulate the fusion of the two detectors' results. In section IV, the experiments are presented.

2.0 The Target's Temporal Template Model and GLRT implementation

It is known that the ideal response x of a metallic object in UWB SAR image can be temporarily decomposed into two components: the driven response and the resonance response. The driven response always precedes the resonance response, which only contains damped sinusoids. The total response $\boldsymbol{x}$ can be divided in time as

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_d^T & \boldsymbol{x}_r^T\end{bmatrix}^T \qquad (1)$$

, where $\boldsymbol{x}_d$ is the $N_d \times 1$ driven response, and $\boldsymbol{x}_r$ is $N_r \times 1$ resonance response. Let $\boldsymbol{y}$ be an $N_y \times 1$ column vector in the down-range profile of the UWB SAR image. The measurement vector $\boldsymbol{y}$ is assumed formed by the total response $\boldsymbol{x}$ and a Gaussian noise vector $\boldsymbol{w}$. Obviously, $N_y = N_d + N_r$, where N_d and N_r are usually target dependent. Now, $\boldsymbol{y}$ can be written as

$$\boldsymbol{y} = \begin{bmatrix}\boldsymbol{y}_d \\ \boldsymbol{y}_r\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_d \\ \boldsymbol{x}_r\end{bmatrix} + \begin{bmatrix}\boldsymbol{w}_d \\ \boldsymbol{w}_r\end{bmatrix} \qquad (2)$$

,where $\boldsymbol{w}_r$ and $\boldsymbol{w}_d$ are the corresponding noise vectors. It is important to note that for foliage or non-metallic objects only the driven response $\boldsymbol{y}_d$ exists. The energy contained in the driven response is much larger than that contained in the resonance response, which hinders the design of detectors based on the resonant response.

Let's assume that the ideal resonance response $\boldsymbol{x}_r$ belongs to a known M_r -dimensional orthogonal signal subspace represented by a $N_r \times M_r$ matrix, $\boldsymbol{L}_r$. If we apply linear transform $\boldsymbol{L}_r^T$ to the $\boldsymbol{y}_r$, then we get

$$\boldsymbol{z}_r = \boldsymbol{L}_r^T \boldsymbol{y}_r = \boldsymbol{S}_r \boldsymbol{a}_r + \boldsymbol{v}_r \qquad (3)$$

, where $\boldsymbol{a}_r$ is the representation vector, and $\boldsymbol{S}_r$ a $M_r \times M_r$ matrix of rank K_r describing the locations of known components. At most one element in each row and each column would be equal to one, and the remaining of the elements are zero. $\boldsymbol{v}_r = \boldsymbol{L}_r^T \boldsymbol{w}_r$ is the colored Gaussian noise vector with zero mean and covariance $\sigma_r^2 \boldsymbol{L}_r^T \boldsymbol{L}_r = \sigma_r^2 \boldsymbol{Q}_r$.

Following the approach in [Yen and Principe, 1997a], we can show that the GLRT testing statistics for this problem is given by

$$t_r = \frac{\boldsymbol{z}_r^T \boldsymbol{Q}_r^{-1} \boldsymbol{z}_r - \boldsymbol{z}_r^{c^T} \boldsymbol{Q}_r^{c^{-1}} \boldsymbol{z}_r^c}{\sigma_r^2} \underset{\mathbf{H}_0}{\overset{\mathbf{H}_1}{\gtrless}} T_r \qquad (4)$$

, where T_r is the threshold for t_r. $\boldsymbol{Q}_r^c$ is the correlation for the null space of the resonant response (dimension $M_r - K_r$). In our UWB SAR scenario, we assume σ_r^2 is unknown, and can be estimated from the neighboring $p \times 1$ sample vector $\boldsymbol{u}$ by $\hat{\sigma}_r^2 = (\boldsymbol{u}^T\boldsymbol{u})/p$. It can be shown [Yen and Principe, 1997a] that the tested statistics t is F-distributed, and the CFAR property can be achieved. Similarly, for the driven response, the GLRT testing statistics can be written

$$t_d = \frac{\boldsymbol{z}_d^T \boldsymbol{Q}_d^{-1} \boldsymbol{z}_d - \boldsymbol{z}_d^{c^T} \boldsymbol{Q}_d^{c^{-1}} \boldsymbol{z}_d^c}{\sigma_d^2} \underset{\mathbf{H}_0}{\overset{\mathbf{H}_1}{\gtrless}} T_a \qquad (5)$$

, where T_d is the threshold.

The accuracy of the GLRT is highly dependent upon the subspace utilized to represent the signal. The most compact signal representation is obtained when the basis are the eigenfunctions of the signal we are dealing with. Due to the fact that the resonant response is a linear combination of decaying sinusoids, we proposed to utilize the Laguerre functions to implement the bases of the projection space [Yen and Principe, 1997a]. The k-th Laguerre sequence $l_i(n, u)$ is obtained by applying the Gram-Schmidt orthogonalization to the following exponential sequence [Mahammad and Ahmed, 1991]

$$f_i(n, u) = n^k u^n \qquad (6)$$

The Laguerre sequences are a complete set of l^2 [Mahammad and Ahmed, 1991] and their Z transform display the following recursive relation

$$\begin{aligned} L_{i+1}(z, u) &= L(z, u)L_i(z, u) \\ L(z, u) &= \frac{z^{-1} - u}{1 - uz^{-1}} \end{aligned} \qquad (7)$$

An added advantage of the Laguerre functions is that they are recursively computable, yielding very compact and computationally efficient algorithms (O(K)), where K is the number of basis (even better that the computational complexity of the FFT).

The implementation of our detector based on the GLRT is shown in Figure 1. The down range profile is sent to two Laguerre delay lines that implement the subspace. Each tap output is a projection on a basis. In order to implement (4) and (5), one has to select what are the taps that contain most of the information about the resonant response (and the driven response). These taps constitute the signal subspace to implement the GLRT. The remaining taps represent the null space for the signal.

3.0 Temporal Detection Fusion Using a Neural Network

The sequential detection fusion can be viewed as a two-hypothesis detection problem with two individual detectors for the driven and resonance responses, respectively. It can be shown [Chair and Varshney, 1996] that the optimum decision rule to fuse the detection results t_d and t_r is given by

$$f(u_1, u_2) = sign(a_0 + a_1 u_1 + a_2 u_2) \quad (8)$$

where

$$\begin{aligned} u_1 &= sign(t_d - T_d) \\ u_2 &= sign(t_r - T_r) \end{aligned} \quad (9)$$

The optimum weights are given by

$$a_0 = \log(P_1/P_0) \quad (10)$$

and

$$\begin{aligned} a_i &= \log\frac{1-P_{M_i}}{P_{F_i}} \qquad if\ u_i = \ +1 \\ a_i &= \log\frac{1-P_{F_i}}{P_{M_i}} \qquad if\ u_i = \ -1 \end{aligned} \quad (11)$$

, where P_{M_i} and P_{F_i} are the miss detection probability and false alarm probability for the $i\ th$ detector, respectively. However, the above optimal fusion rule implies fixed local detectors with preset thresholds. Moreover all the weights are pre-calculated based on the theoretical signal distributions which is unrealistic for UWB SAR. The neural network approach we propose extends the previous fusion rule in (8) with the hyperbolic tangent function instead of the hard-limit sign function yielding

$$f'(u_1', u_2') = \tanh(a_0' + a_1' u_1' + a_2' u_2') \quad (12)$$

, where

$$\begin{aligned} u_1' &= \tanh(t_d - T_d) \\ u_2' &= \tanh(t_r - T_r) \end{aligned}, \quad (13)$$

The advantage is that all the weights and the thresholds can be adapted using backpropagation [Haykin, 1994] to give better performance. The overall optimum detection scheme can be implemented as shown in Fig. 1.

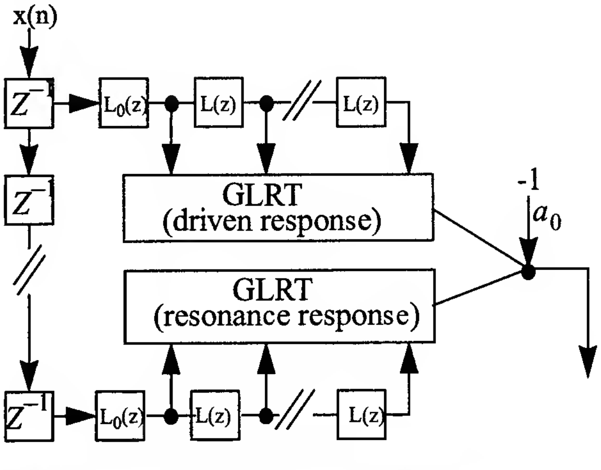


Figure 1. GLRT with Laguerre Network

4.0 Simulation results

We will show that the subspace detection scheme presented in [Yen and Principe, 1997a] can be largely improved using the sequential detection. The data for the simulation provided by ARL's UWB [McCorkle and Nguyen, 1994] is a 128x5376 image. That image contains a vehicle target around sample 3,000, and foliage along down range cell over 4,000. We use a 10 dimensional projection space implemented by a 10th order Laguerre delay line with the recursive parameter μ equal to 0.4. Our previous results show that the resonance response is within the subspace expanded by three Laguerre kernels with order equal to 5, 6 and 7. The remaining taps constitute the signal null space.

The detection statistics of the usual GLRT based on the 1D resonance model implemented by (4) or (5) is shown in Fig 2 (first 3,600 down range cells) and Fig 3 (remaining down range cells). Although the parameter μ is not optimized and only three out of the ten taps are used to define the signal subspace, the algorithm is able to detect the targets around sample 3000 and provide a low output for the clutter. However, there is also a false alarm around 4,000 with a response as high as the target.

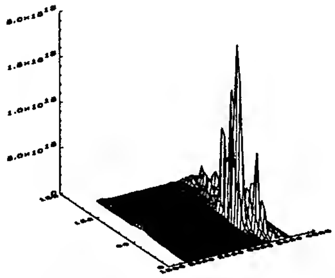


Fig. 2 the detection statistics based on the GLRT of the 1D resonance model along the down range cells 1801~3600

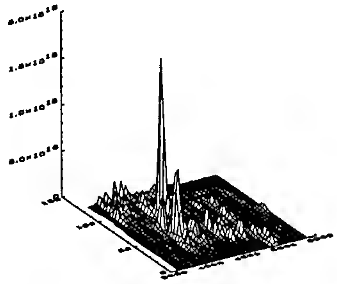


Fig. 3 the detection statistics based on the GLRT of the 1D resonance model along the down range cells 3601~5376

The detection statistics of the GLRT based on the sequential fusion scheme are shown in Fig 4 and Fig 5, respectively. Comparing the detection statistics of target in Fig 3 with that of the foliage in Fig 5, we can see that the foliage detection is totally suppressed, and that there are more detections corresponding to the accurate target locations.

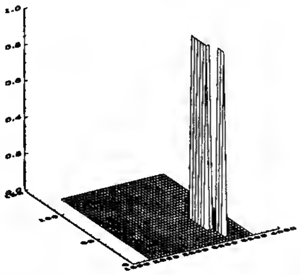


Fig. 4 the detection statistics based on the fused detector along the down range cells 1801~3600

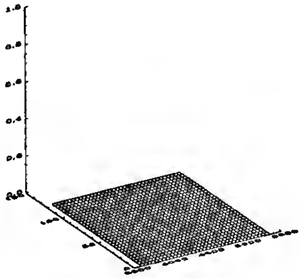


Fig. 5 the detection statistics based on the fused detector along the down range cells 3601~5376

5. Conclusion

Although these are preliminary tests, the idea of exploiting the structure of the UWB response from metallic objects by fusing the driven response with the resonant response seems to improve the accuracy of the focus of attention. In the future work, we will be carefully designing the projection space (adapting the parameter μ) to extract most of the resonance response, and making optimal decisions for the search of the signal and noise spaces. We are researching other kernels that match even better the characteristics of the driven response. One aspect that we would like to mention is the simplicity of this implementation that can lead to on-line algorithms for focus of attention in UWB.

Acknowldgment

This work was partially supported by the grant ONR N00014-94-1-0858.

References

L. K. Yen, J. C. Principe, "Adaptive target detection using Laguerre networks," to appear in ICNN97, Houston

L. K. Yen, and J. C. Principe, "Target detection in UWB SAR using Laguerre networks with spatial templates," to appear in SPIE97, Orlando.

Y. Wang, R. Chellappa, and Z. Qinfen, "Detection of point targets in high resolution synthetic aperture radar images," ICASSP v 5 1994. p V-9-12

J. Li and E. Zelnio, "Target detection with synthetic Aperture Radar," IEEE Transactions on Aerospace and Electronic Systems, vol 32, no. 2, pp. 613-627, Apr. 1996

Z. Chair, and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," IEEE Transactions on Aerospace and Electronic Systems, AES-22, no. 1, pp. 98-101, Jan. 199

N. Ansari, E. S. H. Hou, B. Zhu, and J. G. Chen, "Adaptive Fusion by Reinforcement learning for distributed detection systems," IEEE Transactions on AerospaceandElectronicSystems,AES-32,no.2,pp. 524-530, Apr. 1996

M. A. Mahammad and N. Ahmed, "Optimal Laguerre networks for a class of discrete-time systems," IEEE Tran. on Signal Processing, vol. 39, no. 9, pp.2104-2108, Sep. 1991

L. L. Scharf, "Statistical Signal Processing, "Addison-Wesley.

W. Chang, and M. Kam, "Asynchronous distributed detection," IEEE Transactions on Aerospace and Electronic Systems v 30 n3 Jul 1994. p 818-826

R. Srinivasan, "Designing distributed detection systems," IEE Proceedings, Part F: Radar and Signal Processing v 140 n3 Jun 1993. p 191-197

S. Haykin, "Neural networkss,"Macmillan 1994

M. McCorkle and L. Nguyen, "Ultra wideband bandwidth synthetic aperture radar focusing of dispersive targets," Technical Report ARL-TR-305, Army Research Laboratory, Adelphi, M.D., Apr. 1994

A Self-organizing Principle for Segmenting and Super-resolving ISAR Images

Frank M. Candocia
candocia@cnel.ufl.edu

Jose C. Principe
principe@cnel.ufl.edu

Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL. 32611
http://www.cnel.ufl.edu

Abstract

We present and illustrate the use of a bottleneck system for the segmentation and super-resolution of ISAR targets. The system is shown to be comprised of three basic subsystems: a compressing transformation, a bottleneck processor, and a decompressing transformation. We describe each subsystem and discuss the processing responsible for segmentation and super-resolution within this framework. Results using this network are assessed and issues regarding performance are introduced.

1. Introduction

Feature extraction is critical in many signal and image processing applications but our ability to automatically extract features from data is very limited. In preselecting features, we rely too much and too often on our apriori knowledge of the problem. This methodology can be problematic when such apriori knowledge is scarce and also hinders our ability to quantify the quality of the features chosen. In our opinion, feature extraction should be based on self-organizing methods because the signal's samples are the only available information source. Such a methodology is encountered in prediction where the input signal is cleverly utilized as a desired response.

Prediction is a way of self-organizing a system for time signals but it is much more difficult to apply for images. The other known principle of self-organization with an implicit desired response is auto-association with a bottleneck layer [Bourland and Kamp, 1988]. This can be thought of as the equivalent to prediction for images and also gives us a model for the intrinsic structure of the data. In essence, we seek to model image data for a given class of imagery that is "independent" of its scale. Such an approach has been proposed in [Candocia and Principe, 1997].

The idea of bottleneck processing is not new. This type of processing has had much success in the areas of image compression [Jain, 1989] and subspace pattern recognition [Oja, 1983]. In image compression, the saving of a few transform coefficients to represent data in a compressed form can be formulated as a bottleneck process. In subspace pattern recognition, the reduction in dimensionality of a signal is an important practical step to obtaining discriminant functions. This type of processing, though, is not restricted to auto-association. It has seen use in hetero-association via non-symmetric PCA [Kung, 1993]. The work presented here makes use of such processing in a more general, non-traditional fashion.

2. The Bottleneck System

The bottleneck system (BNS) as an auto-associator is composed of three basic components: (1) a compressing transformation which is responsible for producing a reduced representation of the input signal, (2) a bottleneck processor which further processes the compressed signal space and (3) a decompressing transformation which is responsible for reconstructing the input signal. This is illustrated in fig. 1.

The input to the BNS is given by the vector x and the reconstructed output is denoted $\hat{x}$. The first and last blocks of this processing are the projections that constitute the forward and inverse transforms of a signal, respectively. These transforms need *not* be linear. The only constraint is that the dimensionality of the input space be reduced. The compressed input space is denoted y and this feeds into the bottleneck processor (BNP). The BNP generates information z regarding y which could aid in the reconstruction of x; it could also receive additional information q as input to aid in the generation of z. The information z and compressed input space y then feed into the decompressor.

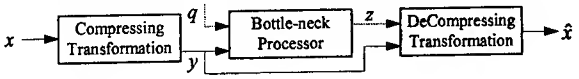


Figure 1. Block diagram of the BNS.

Commonly used compression techniques make use of linear transforms in the first and last blocks of the BNS and the BNP is simply an identity transformer $f(y)=y$ with no q or z. In this case a vector $x\in\Re^N$ is transformed to a vector $y\in\Re^M$ via $y=Wx$ where $M < N$. The approximate reconstruction of vector x is given by $\hat{x}= W^H y$ where H denotes the Hermitian of W. In PCA, it is known that W is a matrix whose rows are the M largest eigen-valued eigenvectors of x's covariance matrix and y is a vector of the principal components of x. In transforms such as the DCT or DFT, the rows of W constitute sinusoidal basis functions (real and complex, respectively) for which to project x onto and y are the corresponding transform coefficients. The basis functions used for projecting are the ones that yield the largest transform coefficients.

3. Pre-processing the Input

This paper addresses the ability of a BNS to (1) segment target vs. background and (2) super-resolve a 1m × 1m resolution ISAR data set to 1ft × 1ft resolution. These two problems are very different. The first is a clustering (classification) problem and the second is a regression problem. To aid in tackling these problems with the BNS, it is important to consider what pre-processing of the input, if any, should be performed. The pre-processing step should reflect any characteristic of the data inherent to alleviating the complexity associated with the problem. Let us now note that any reference to an ISAR image is referring to the PWF transformed ISAR target. An ISAR image is thus real valued. The 1ft × 1ft resolution training and test images are illustrated in fig. 2.

It is evident that segmenting a target versus background in an ISAR image requires information about the brightness of a pixel and eventually texture. Here we will work simply with brightness. This brightness is directly proportional to the amount of backscatter received by the radar - which is usually large for metallic objects relative to non-metallic ones. As such, the segmentation problem is local in nature and should make use of a local brightness measure. This is done by transforming local neighborhoods of our ISAR images into spherical coordinates. More specifically, each $H \times H$ neighborhood of our ISAR images is regarded as a vector (in Cartesian coordinates) in an H^2 dimensional vector space. Our set of vectors, or ISAR image neighborhoods, are then transformed to their multi-dimensional spherical coordinates. One of the coordinates in this representation is known to be the length or norm of the vector. This quantity is descriptive of the brightness of an ISAR image neighborhood. This is the preprocessing performed with regards to our segmentation problem.

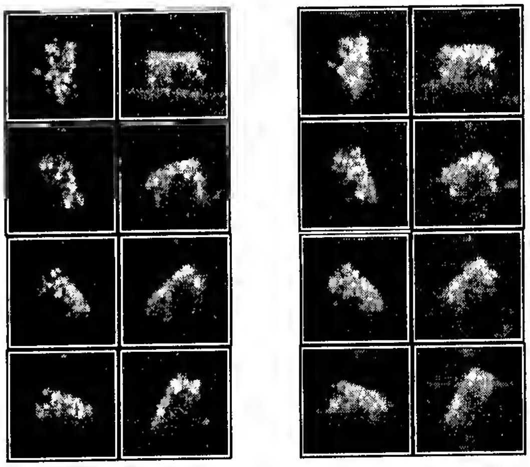


Figure 2. High resolution ISAR images. (left 8) training (right 8) testing.

The preprocessing for super-resolving an ISAR image is different from that just described. The backscatter at various points on targets can be quite similar - even across a set of different targets situated at varying aspect angles relative to the radar. This also suggests a local approach to the super-resolution problem. What is not clear at present is which set of descriptors (and for that matter, pre-processor) retains the most information about a class of images across resolutions. In our pre-processing, we have decided to normalize each vector (ISAR image neighborhood) to unit length. The effect of this operation will be discussed in the next section.

4. Defining the Blocks of the BNS

The framework for the processing of ISAR images is given by the BNS. Here we motivate and define the processes contained within each of the blocks pictured in fig. 1.

4.1 The Compressing Transformation

The process of super-resolving ISAR/SAR information involves the increase of its resolution. This process is akin to that of interpolation in images. Here we synthesize the lower resolution 1m × 1m ISAR set to be super-resolved by decimating our original 1ft × 1ft ISAR images by a factor of 3 × 3. These images are illustrated in fig. 3. Note that now we have two versions of the same imagery with different resolutions to train a model. Later on, the model can be used on new low resolution data to enhance it.

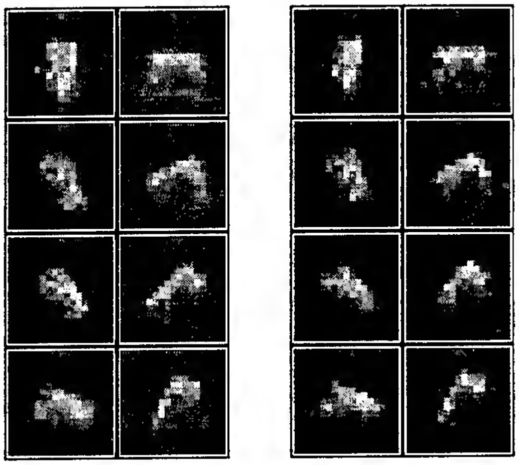


Figure 3. Synthesized low resolution ISAR images. (left 8) training (right 8) testing.

Decimation is the process of appropriate lowpass filtering followed by subsampling [Crochiere and Rabiner, 1981]. Notice that this is a non-invertible, non-linear transformation which yields a coarse representation of our input images. Neighborhoods are then extracted from the decimated images. The set of these neighborhoods will be denoted by $y = \{y_p\}$ where each y_p is a distinct $H \times H$ neighborhood from the compressed training images that has been converted to a vector by stacking the columns of the square neighborhood (or matrix) one on top of the other. These neighborhoods are samples of the compressed input space y alluded to in fig. 1. Our compressing transformation is thus a decimation by a factor of 3 × 3 followed by an extraction of neighborhoods from the resulting images. This compressing transformation also works well for the segmentation problem. It further reduces speckle in our coarser ISAR images due to the decimation (albeit at the expense of image detail). However, the detail is not significant to this segmentation problem.

4.2 The Bottleneck Processor

It is important to be able to extract features (in a self organizing manner) from the compressed and pre-processed input space y. These features serve to establish the M most relevant descriptors of this space and are subsequently used to partition it. Here, the feature extraction is accomplished via vector quantizion (VQ) of the neighborhoods y_p of the compressed and pre-processed training images in set y. A number of VQ algorithms exist including k-means, Kohonen's self organizing feature map [Kohonen, 1990] and the neural gas algorithm [Martinez et al., 1993]. The codebook vectors or quantization nodes q_z, $z = 1, \cdots, M$ that result from VQ are the intrinsic descriptors of y. We denote the set of quantization nodes by q, i.e. $q=\{q_1,\ldots,q_M\}$ and each $q_z \in \Re^K$ where $K=H^2$.

Our study makes use of the BNP illustrated in fig. 4. There are two separate inputs to this block: q and y as previously discussed. Clustering neighborhoods y_p based on closest distance to each q_z results in a hard partitioning of y into regions that are most correlated. Specifically, the cluster C_z contains those neighborhoods y_p of y that are closest to q_z in Euclidean distance. This is given in eqn. (1).

$$C_z = \left\{ y_p : \left\| q_z - y_p \right\|_2 < \left\| q_a - y_p \right\|_2 \right\} \qquad (1)$$

where $z = 1, \cdots, M$ and $z \neq a$. The single integer output $z \in \{1, \cdots, M\}$ of the BNP represents the cluster C_z that a neighborhood y_p belongs to.

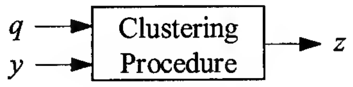


Figure 4. The BNP implemented for this paper.

The segmentation problem mentioned needs only M=2 quantization nodes. One node theoretically clusters neighborhoods corresponding to targets and the other corresponds to non-target neighborhoods (no shadows are considered).

The super-resolution approach makes use of M=30 quantization nodes, i.e. the neighborhoods y_p are partitioned into 30 clusters which will each be super-resolved. Note that (the vector for) each y_p clustered is of unit length due to the pre-processing that was performed. This form of pre-processing yields scale invariant neighborhoods, i.e. for two neighborhoods a and b, if $a \approx kb$[1] (k a scalar), these two neighborhoods have the same underlying reflectance properties, regardless of the illumination in the scene. This is an assumption made in homomorphic image processing [Dony and Haykin, 1995] which may not be valid for our ISAR images (as alluded to earlier). The question as to what pre-processor to use for super-resolution needs further addressing.

4.3 The DeCompressing Transformation

The decompressing transformation is not needed for the segmentation problem; the output z of the BNP describes the cluster a neighborhood corresponds to. Each neighborhood in the ISAR image is thus assigned to the target or non-target cluster.

The decompressing transformation is obviously needed for the super-resolving of ISAR images. The neighborhoods y_p have been clustered into M=30 groups. Our decompressing transformation is composed of M=30 individual affine transformations $\{W_z, B_z\}$ - each tailored to the specific information contained in cluster C_z, $z = 1, \cdots, M$. z now is essentially a "pointer" or indicator as to which individual transformation to use. The reconstruction of a neighborhood $\hat{x}_p$ is accomplished by:

$$\hat{x}_p = \|y_p\|_2 \cdot uvec\left(W_z y_p + B_z\right) \; ; \quad y_p \in C_z \qquad (2)$$

where W_z and B_z are the weight matrix and bias vector associated with an affine transformation, $uvec(\cdot)$ undoes the vectorizing operation that was performed to the neighborhoods y_p in set y and the 2-norm of y_p is used to restore the length of the vector which was removed during pre-processing. Details concerning the individual transformations are discussed in [Candocia and Principe, 1997].

5. Results

The results presented here utilized ISAR targets that were PWF transformed from the TABILS 24 data set. The resolution of this data set is 1ft × 1ft. We chose 8 targets for each of our training and test sets spanning 180° of aspect angles. The difference between target aspect angles in each set was 22.5°. The corresponding 1m × 1m low resolution training/test data was simulated through decimation of the high resolution training/test data as discussed earlier. Neighborhoods of 5 × 5 (H=5) were utilized in the extraction of features both for the segmentation and super-resolution examples. The features automatically found through clustering the low resolution neighborhoods for the purpose of target segmentation are illustrated in fig. 5.

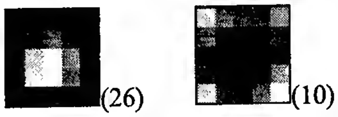


Figure 5. Features extracted for target segmentation. (left) target, (right) non-target.

These features have been scaled to visually enhance the structure associated with each feature. The number in parenthesis indicates the 8-bit gray level difference between the brightest and darkest value in each feature. Notice that the extracted feature corresponding to targets has a peaky center. This is consistent with the notion that ISAR targets are characterized by bright point scatters. The non-target feature is, interestingly enough, an "anti-target" feature. It characterizes local information that is "opposite" that of target information. Fig. 6 illustrates the target vs. non-target segmentation results.

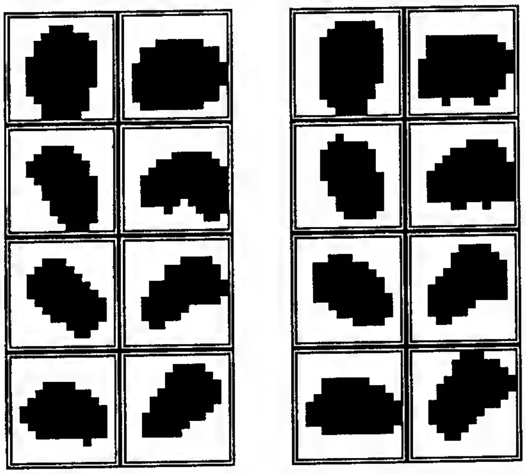


Figure 6. Segmented low resolution images. (left 8) training (right 8) test.

[1] $a \approx kb$ is our short hand notation for $\|a - kb\|_2 < \varepsilon$ where $\varepsilon > 0$ and small and a,b are vectors

It is important to note that pre-processing is critical to the types of features extracted with the self-organized clustering scheme. The capacity of the network to interpolate or super-resolve the low resolution representations to the resolution of the original images is illustrated in fig. 7.

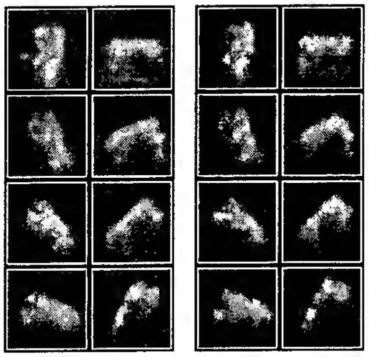


Figure 7. Super-resolved images. (left 8) training (right 8) test.

The pre-processing of the low resolution images consisted of a simple normalization of the image neighborhoods. Here, M=30 features were extracted from the pre-processed low resolution images. Note that we are attempting to establish a system that recovers 9 times the information presented to it. The cropping effect is due to not super-resolving image portions with low "confidence". This confidence is directly attributed to the amount of available data about the location to super-resolve.

The test portion of fig. 3 with fig. 7 shows that the BNS approach is capable of capturing the salient characteristics of a class of images across resolutions.

6. Discussion and Conclusions

The self-organization approach solved the segmentation problem here with little effort. In order to segment, it was important to have a local measure of brightness available to the clustering. In fact, the coordinate largely responsible for the segmenting was the length coordinate in the spherical coordinate transformation utilized. By clustering on this sole coordinate, comparable results to those of fig. 6 were obtained.

The bottleneck approach extracts and models the image structure across resolutions in the image set. The derived model can then be applied to new low resolution images to super-resolve them. For instance, a 1m × 1m PWF radar image of targets could be digitally interpolated on the fly to 1ft × 1ft using our method.

We are still investigating many issues concerning the ISAR/SAR super-resolution problem. Our research on optical images has shown the existence of highly correlated information across scales and that this information can be exploited for interpolation. There is an analogous relation to this in the electro-magnetic domain of SAR which we also wish to exploit. Very probably, the super-resolution should be performed both at the complex and PWF transformed image levels. Also, questions as to hard vs. soft partitioning of the low resolution space are being examined as well as what pre-processing is "most appropriate" for the super-resolution problem.

Acknowledgements

This work was partially supported by DARPA F33615-97-1019.

References

Bourland H. and Kamp Y. (1988). "Auto-association by the multilayer perceptron and singular value decomposition", *Biological Cybernetics*, Vol. 59, pp. 291-294.

Candocia F.M. and Principe J.C. (1997). "A Neural Implementation of Interpolation with a Family of Kernels", To appear in *Proc. Int. Conf. Neur. Net.* (ICNN 97), Houston, Tx.

Crochiere R.E. and Rabiner L.R. (1981). "Interpolation and Decimation of Digital Signals - A Tutorial Review", *Proc. of IEEE*, Vol. 69, No. 3, pp. 300-331.

Dony R.D. and Haykin S. (1995). "Optimally Adaptive Transform Coding", *IEEE Trans. Image Proc.*, Vol. 4, No. 10, pp. 1358-1370.

Jain A.K. (1989), Fundamentals of Digital Image Processing, Englewood Cliffs, NJ: Prentice Hall.

Kohonen T. (1990). "The Self-organizing Map", *Proc. IEEE*, Vol. 78, pp. 1464-1480.

Kung S.Y. (1993). Digital Neural Networks, Englewood Cliffs, Prentice Hall, NJ: Chap. 8.

Martinez T.M., Berkovich S.G. and Schulten K.J. (1993), "Neural-gas' network for vector quantization and its application to time-series prediction", *IEEE Trans. Neur. Net.*, Vol. 4, No. 4, pp. 558-569.

Oja E. (1983), Subspace Methods of Pattern Recognition, Letchworth, UK: Research Studies Press.

Matching of Articulated Objects in SAR Images

Joon Soo Ahn and Bir Bhanu*
College of Engineering, University of California, Riverside, CA 92521-0425
E-Mail: {ahn, bhanu}@constitution.ucr.edu
URL: http://constitution.ucr.edu

Abstract

We present distance weighted geometric hashing-based indexing technique and a matching technique using body models and turret models for the automatic target recognition of articulated objects in synthetic aperture radar images. For each target, 360 body models and 360 turret models are built. These models are independent of the relative position between the body and turret. Four non-articulated targets (SCUD missile launcher, T-72 tank, M1 tank and T-80 tank) are used in the indexing stage to build the look-up table. In the matching stage, M1 tanks with turret rotated 30°, 60°, 90° relative to the body are used as data.

1 Introduction

Recognition of articulated objects in SAR images is a challenging problem. A simple approach may consider each of the articulated parts of an object as separate objects. However, such an approach is quite inefficient since it will require a large model database. We want to develop an efficient recognition approach that inherently models the articulated nature of an object such as a SCUD missile launcher or a tank with different positions of its turret.

Some of the representative work for target recognition using SAR images includes [2], [4] and [5]. These papers focus on template matching techniques in which the templates are manually designed. Recent work on the recognition of articulated objects in SAR images includes [1], [3]. In this paper we describe a geometric hashing-based indexing with weighted voting and a matching technique using body models and turret models. We have evaluated the performance of our initial approach using XPATCH data.

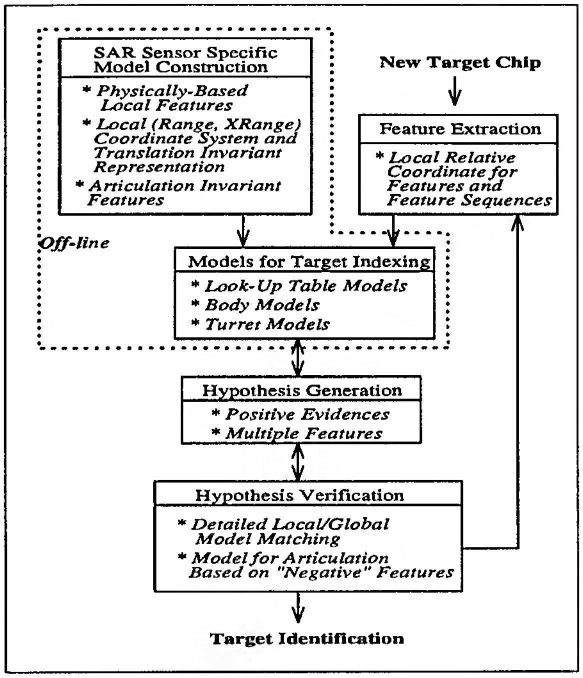

SAR Sensor Specific Model Construction
* Physically-Based Local Features
* Local (Range, XRange) Coordinate System and Translation Invariant Representation
* Articulation Invariant Features
Off-line
New Target Chip
Feature Extraction
* Local Relative Coordinate for Features and Feature Sequences
Models for Target Indexing
* Look-Up Table Models
* Body Models
* Turret Models
Hypothesis Generation
* Positive Evidences
* Multiple Features
Hypothesis Verification
* Detailed Local/Global Model Matching
* Model for Articulation Based on "Negative" Features
Target Identification

Figure 1: System for recognizing articulated targets in SAR images.

1.1 Approach

Figure 1 shows the system for recognizing articulated targets in SAR images. Our approach is based on local features and local reference coordinate system. The models for the look-up table are constructed by extracting the relative positions of the features from the non-articulated training data. The body and turret models are constructed by using three different articulation configurations of tank targets.

Detailed description of the geometric hashing technique, specifically designed for SAR, using the look-up table is given in [3]. Distance weighted geometric hashing-based indexing is an enhancement of the basic geometric hashing technique which increments each vote not by one, but by $max(|dx|, |dy|)$ where $|dx|$ and

*This work is supported by grant MDA972-93-1-0010. The contents and information do not reflect the position or policy of the U.S. Government.

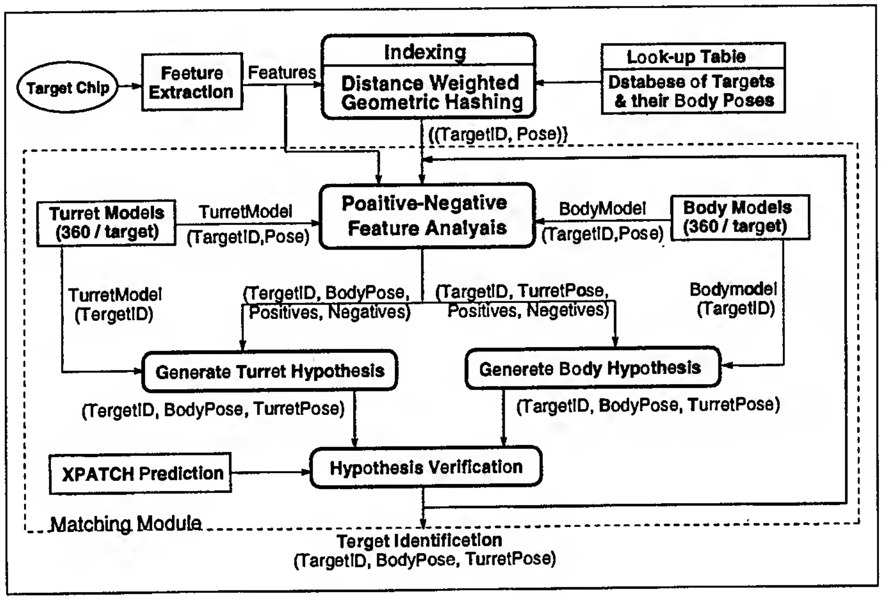

Target Chip
Feeture Extraction
Features
Indexing
Distance Weighted Geometric Hashing
Look-up Table
Dstabese of Targets & their Body Poses
((TargetID, Pose))
Turret Models (360 / target)
TurretModel
(TargetID,Pose)
Positive-Negative Feature Analyais
BodyModel
(TargetID,Pose)
Body Models (360 / target)
TurretModel (TergetID)
(TergetID, BodyPose, Positives, Negatives)
(TargetID, TurretPose, Positives, Negetives)
Bodymodel (TargetID)
Generate Turret Hypothesis
Generete Body Hypothesis
(TergetID, BodyPose, TurretPose)
(TargetID, BodyPose, TurretPose)
XPATCH Prediction
Hypothesis Verification
Matching Module
Terget Identificetion
(TargetID, BodyPose, TurretPose)

Figure 2: Indexing and matching components for recognizing articulated targets.

$|dy|$ represents the absolute values of the relative distances between two peak features in the direction of range and cross-range, respectively. This module generates set of hypotheses with target identification and pose which may be the body pose or the turret pose.

For the matching process, 360 body models and 360 turret models are built (one/degree azimuth) for each tank target and used to find the positive and negative point features from the data (see Figure 2). The positive features are used to generate hypotheses for target identification and body pose. Negative features are used to generate hypotheses for turret pose. As there might be multiple hypotheses from the indexing module, the matching module will loop for each of the hypotheses.

The basic assumption is that the positive points are from the body part. The negative features are produced as a result of articulation and interaction between the body and the turret. For some targets like the M1 tank, the turret part is so large that the indexed pose may be the turret pose. To resolve this problem, the **Positive-Negative Feature Analysis** stage uses body models and turret models to detect the part. This stage uses the body model with specific target type (ID) and body pose to detect the positive features. If the number of positive points are larger than a fraction of the number of the specific body model points, then we generate the turret hypothesis. If the number of positive points are less than a fraction of the number of the body model points, we use the turret model for the target ID and turret pose and go to the next step to generate the body hypothesis.

2 Off-line Model Building

2.1 Extraction of scattering centers to build non-articulated model base

We employed a simple method of detecting local maxima. The method is based on comparing the pixel value with its immediate eight neighbors. If the current pixel value is greater than all the other immediate eight neighbors, then it is a local maximum.

2.2 Building non-articulated model base

We extracted the top fifty local maxima from the images of SCUD missile launcher with missile down, T72 tank, M1 tank and T80 tank with turrets straight to the bodies. The top fifty local maxima are then sorted in descending order of their magnitudes of SAR return signals.

```
build_non-articulated_model_base()
{  $N$ = number of local maxima
   for (Object = 1 to NO_OF_OBJECTS){
      for (Angle = 0 to 359){
         model = get_model_image(Object, Angle)
         peaks = extract_local_maxima(model, $N$)
         save(Object, Angle, peaks)
}}}
```

2.3 Building body models

Figure 3 shows T-72 tanks with turret 0°, 60°, and 90° rotated relative to the body whose pose is 283°. The fourth figure shows the body model of T-72 tank

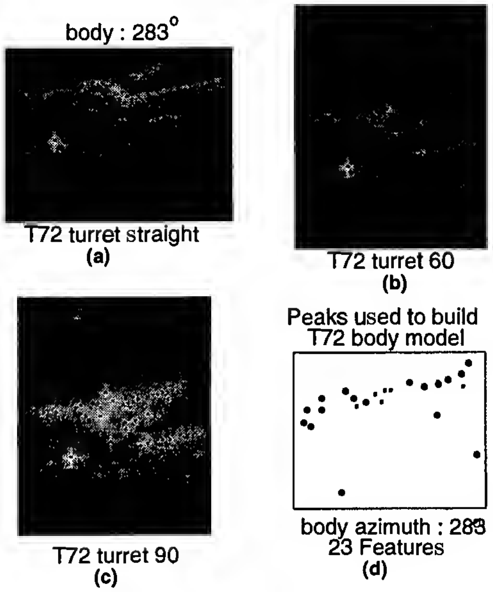


Figure 3: T-72 body model with azimuth 283°. (a) body azimuth = 283°, turret straight. (b) body azimuth = 283°, turret 60°. (c) body azimuth = 283°, turret 90°. (d) body model at 283°.

with the body azimuth at 283°. To build this model, first, conjunction operation between two images with different turret poses has been performed. This operation generates three sets of point features (0° & 60°, 0° & 90°, 60° & 90°). Then, union operation on these three sets of point features give a set of point features which represent the body model of T-72 at 283°. The conjunction operation represents the best matching between two sets of point features. The union operation represents the union of two sets of point features where one set of point features are translated appropriately to have the best matching between them. The result in 3(d) shows 23 point features. There are 17 large dots in the model, which represents the point features for the best matching among the three sets of original point features.

2.4 Building turret models

Figure 4 shows M-1 tanks with turret 0°, 60°, and 90° rotated relative to the body whose poses are 105°, 45°, and 15° respectively. The fourth figure shows the turret model of M-1 tank with the turret azimuth at 105°. To build this model, first, conjunction operation between two figures with different turret pose has been performed. This operation generates three sets of point features. Then, union operation on these three sets of point features give a set of point features which represent the turret model of M-1 at 105°. The result in Figure 4(d) shows 15 point features. There are 7 large dots in the model, which represents the point features for the best matching among the three sets of original point features.

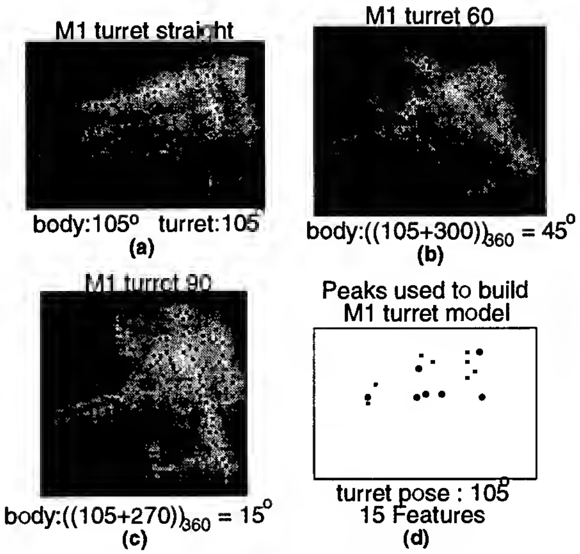


Figure 4: M-1 turret model with azimuth 105°. (a) turret pose = 105°, body azimuth = 105°. (b) turret pose = 105°, body azimuth = 55°. (c) turret pose = 105°, body azimuth = 15°. (d) turret model at 105°.

3 On-line Indexing and Matching

3.1 Distance Weighted Geometric Hashing-Based Indexing

The paper by Jones & Bhanu [3] describes the geometric hashing technique in detail. We have enhanced the indexing module by incrementing the variable *vote* by $max(|dx|, |dy|)$ instead of incrementing it by one. This new weighted voting scheme is different from the original non-weighted voting scheme in employing the relative distance between two points as the weighting factor. This approach improves the indexing results as shown in Figure 5.

3.2 Matching

Following algorithm generates set of hypotheses and finds the best correspondence between data and the set of hypotheses.

```
Exact_Matching(data_image)
{  data = extract_local_maxima(data_image, $N$)
   candidates = Weighted_Geometric_Hashing(data)
   for (each model in top K candidates){
       Positive_Negative(data_points, model_points)
   }
   Sort the hypotheses in descending order of
       the positive points
   Select model with the most matching points
}
```

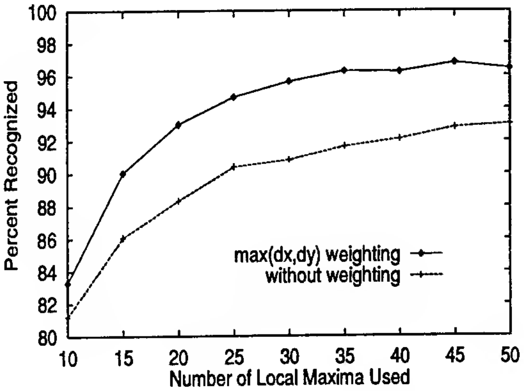


Figure 5: Comparison of the two indexing schemes $max(|dx|, |dy|)$ weighting scheme and no-weighting scheme for the top hypotheses.

Given two sets of points, transform the first set to find the maximum number of corresponding points. In this transform, only translation is considered because the rotation and scaling are taken care of by the design of the recognition system and the peculiar characteristics of the SAR sensor.

Algorithm given in Figure 6 shows how to find the positive and negative points. The best case time complexity is $O(MD)$ and the worst case time complexity is $O(MD^2)$ where M and D represents the number of model and data point features respectively.

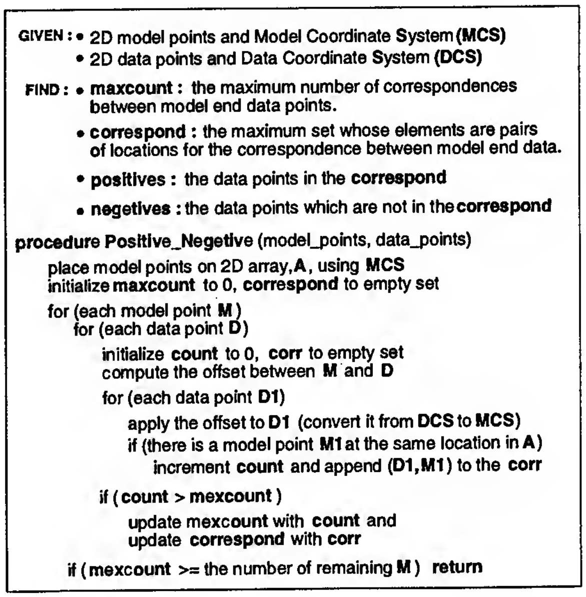
4 Experiment Results

4.1 Building Models

In model building, we use four non-articulated objects, SCUD missile launcher with missile down, T72 tank, M1 tank and T80 tank with the turret straight to the body. For each non-articulated object, we generate 360 images (for each degree in azimuth) for a given depression angle of 15°. From each image, we extracted the top 50 scattering centers with their signal returns and locations as point features of the model. So, the total number of models in the model database is 1440 (4 non-articulated objects * 360).

4.2 Generating testing data

For the testing data, we used M-1 tanks with three articulated turret positions, 30°, 60° and 90° rotated relative to the tank body. For each articulated position, we generated 360 images (one for each degree in azimuth) for a given depression angle of 15°. From each image, we extracted the top 50 scattering centers with their signal returns as point features of the data. So, the total number of data in the experiment is 1080 (3 articulated objects * 360).



```
GIVEN : • 2D model points and Model Coordinate System (MCS)
        • 2D data points and Data Coordinate System (DCS)
 FIND : • maxcount : the maximum number of correspondences
          between model end data points.
        • correspond : the maximum set whose elements are pairs
          of locations for the correspondence between model end data.
        • positives : the data points in the correspond
        • negetives : the data points which are not in the correspond
procedure Positive_Negetive (model_points, data_points)
    place model points on 2D array,A, using MCS
    initialize maxcount to 0, correspond to empty set
    for (each model point M )
      for (each data point D)
          initialize count to 0, corr to empty set
          compute the offset between M and D
          for (each data point D1)
              apply the offset to D1 (convert it from DCS to MCS)
              if (there is a model point M1 at the same location in A)
                  increment count and append (D1,M1) to the corr
          if ( count > mexcount )
              update mexcount with count and
              update correspond with corr
      if ( mexcount >= the number of remaining M ) return
```

Figure 6: Algorithm for the Positive-Negative Feature Analysis.

4.3 Example of positives from turret

Figure 7 shows an example of the positive points in the data compared to the non-articulated model which is recognized by the turret pose instead of the body azimuth. Note that the positive feature (Figure 7(c)) matches better with the turret model (Figure 7(e)) then with the body model (Figure 7(d)).

4.4 Discussion

Figure 8 shows the results for indexing. The *enhanced indexing with weight* $max(|dx|, |dy|)$ curve shows the target ID and body pose up to the 40th position in the list of hypotheses sorted in descending order of the vote. The cumulative percentage accuracy up to 40th hypotheses is 92.87%.

The *enhanced indexing with positive-negative* curve shows the cumulative percentage of the correct target ID and body pose up to the 40th position in the list of new hypotheses sorted in descending order of the number of positive features from the **positive-negative feature analysis** without the correction of the confusion between body and turret. This curve shows that the **positive-negative feature analysis** brings the correct answer to the top of the hypotheses list if the correct answer is among the top 40 hypotheses of the indexing result.

The *body detection by positive-negative feature analysis* curve shows the cumulative percentage of the correct target ID and body pose after the correction of the confusion between body and turret using the body models and turret models. Based on the top 40 an-

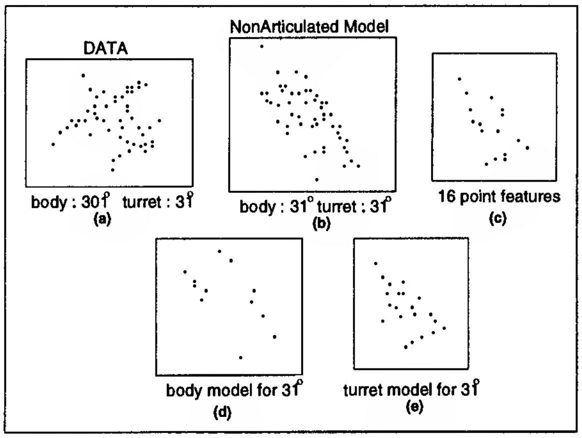


Figure 7: Example of positives from the turret of M-1 tank. (a) Data: body azimuth 301° and turret pose 31°. (b) Non-articulated model hypothesis generated by indexing: body azimuth 31° and turret pose 31°. (c) The positive features of the data detected by the non-articulated model. (d) M-1 body model: azimuth 31°. (e) M-1 turret model: pose 31°.

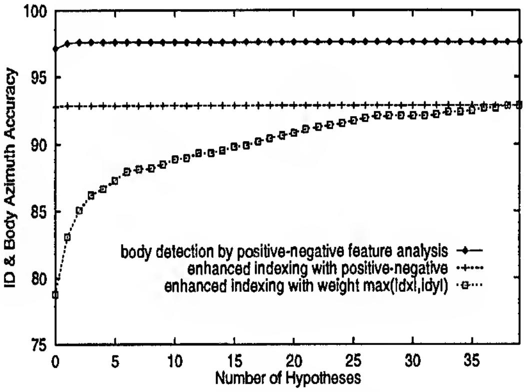


Figure 8: Results for Indexing

swer, the correct target ID and body pose increased to 97.59%, which increases the correct target ID and body pose by 4.74%.

Figure 9 shows the results based on the top hypothesis. In the column of **ID, Body, Turret**, 0, 1 and X represents incorrect, correct and don't care. The column **Exact** shows the result that testing data is recognized at exact pose. The column **Within 5** shows the result that testing data is recognized within +/− 5°.

5 Conclusions and Future Work

In this paper, we have presented the initial research for matching. The goal is to develop physically-based approaches having multiple representations (variety of feature types) for matching to recognize articulated targets in SAR images.

	ID	Body	Turret	Exact	Within ±5°
	0	X	X	0.56(%)	0.56(%)
	1	0	0	1.94(%)	1.39(%)
	1	0	1	0.37(%)	0.19(%)
	1	1	0	46.20(%)	21.02(%)
	1	1	1	50.93(%)	72.69(%)
Exact	99.44(%)	97.13(%)	51.30(%)		
Within ±5°	99.44(%)	98.98(%)	72.87(%)		

Figure 9: Recognition results. These results are based on the top hypothesis only.

We are developing integrated matching technique and analysis for verifying hypothesis (target ID, body pose, turret pose) using articulation variants, positive/negative features, XPATCH prediction, surface reflector type and relative geometry of parts of articulated objects (e.g. M-1 / T-72). We are investigating a Bayesian probabilistic approach to combine the above known information in an integrated manner.

References

[1] B. Bhanu, G. Jones, J. Ahn, M. Li, and J. Yi. Recognition of Articulated Objects in SAR Images. In *Proc. ARPA Image Understanding Workshop*, Palm Springs, California, February 13-16 1996.

[2] D. E. Dudgeon, R. J. Lacoss, C. H. Lazott, and J. G. Verly. Use of persistent scatterers for model-based recognition. In *Proceedings of SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery*, volume 2230, pages 356–368, Orlando, FL, April 1994.

[3] G. Jones and B. Bhanu. Invariant features for the recognition of articulated and occluded objects in SAR images. In *Proc. ARPA Image Understanding Workshop*, New Orleans, LA, 1997. May 13-15.

[4] L. M. Novak, G. J. Owirka, and C. M. Netishen. Performance of a high-resolution polarimetric SAR automatic target recognition system. *The Lincoln Laboratory Journal*, 6(1):11–24, 1993.

[5] A. M. Waxman, M. Seibert, A. M. Bernardon, and D. A. Fay. Neural systems for automatic target learning and recognition. *The Lincoln Laboratory Journal*, 6(1):77–116, 1993.

Lie Group Analysis in Object Recognition

D. Gregory Arnold*
ATR Technology Branch, USAF
WL/AACA Bldg 23, 2010 Fifth Street, WPAFB, OH 45433-7001
E-MAIL: garnold@mbvlab.wpafb.af.mil
HOMEPAGE: http://www.mbvlab.wpafb.af.mil/~garnold/

Kirk Sturtz
Veda, Inc.
5200 Springfield Pike, Suite 200, Dayton, OH 45431-1255
E-MAIL: ksturtz@mbvlab.wpafb.af.mil

Vince Velten
ATR Technology Branch, USAF
WL/AACA Bldg 23, 2010 Fifth Street, WPAFB, OH 45433-7001
E-MAIL: veltenvj@aa.wpafb.af.mil
HOMEPAGE: http://www.mbvlab.wpafb.af.mil/~vvelten/

Abstract

The techniques of Lie group analysis can be used to determine absolute invariant functions which serve as classifier functions in object recognition problems. Lie group analysis is a powerful tool for analyzing complex systems such as the conservation model used in recent thermophysical invariance (TPI) research. We will discuss the mathematics of Lie groups and the application to recognition problems (TPI specifically). The experimental results will demonstrate the validity of the methods and determine the direction of future research. More extensive background and results are available in an extended version of this paper.

1 Introduction

In a nutshell here's what these techniques provide and how they can be used in classifying objects: Lie group analysis will determine if there exists a non-trivial function Φ which assumes a constant value on the set of all roots of an equation $f(\vec{z}) = 0$. The form of the equation remains constant regardless of which particular object we are measuring (viewing), but some of the coefficients in this equation may (and generally will) change depending upon the object being viewed, as for example when $f(\vec{z}) = 0$ expresses a conservation equation. As a result, the set of roots will differ depending upon the object being viewed. Correspondingly the constant value $\Phi(\vec{z})$ will assume a different value depending upon the object being viewed, thus permitting the use of Φ as a classifier function.

In section 2, the mathematics involved are presented, and in section 3 these ideas are applied to the thermophysical invariance problem where the equation $f(\vec{z}) = 0$ is a conservation statement. Finally, some of the theory is confirmed by experimental data and future directions are discussed.

2 Elements of Lie Group Analysis

We explain the theory of Lie Group Analysis as applied to an equation of the form

$$f(\vec{z}) = 0 \tag{1}$$

where $\vec{z} = (z_1, \ldots, z_n) \in \Re^n$ and f is a differentiable function, $f \in C^1(\Re)$. Denote the set of roots of f by

$$V(f) \equiv \{\vec{z} \in \Re^n : f(\vec{z}) = 0\}. \tag{2}$$

If the differential $df \neq 0 \quad \forall \vec{z} \in V(f)$ then f implicitly defines a manifold. We assume this manifold to be connected[1]. Lie group analysis will determine continuous symmetries only; if the manifold is not connected discrete symmetries may exist and cannot be determined by the methods considered here. An example of a discrete symmetry is reflection. In the physical applications we consider in object recognition problems, discrete

*For extended development of the concepts in this paper contact any of the authors.

[1] A manifold M is connected if to each pair of points in M there exists a curve in M connecting the two points.

symmetries are not an issue. The variables under consideration vary continuously.

The concepts and theory given here can be extended to deal with differential equations - and this is where Lie group analysis is used most often. The generalization of these techniques to differential equations is not difficult. See Olver [1993] for such a treatment.

In general, Lie group analysis is applicable for *systems of equations*, however, any system of equations $g_i = 0$ for $i = 1, \dots, m$ can be replaced by a single equation $f \equiv \sum_{i=1}^{m} g_i^2 = 0$ in the sense that $V(g_1, \dots, g_m) = V(f)$. Hence there is no loss of generality in assuming only one equation.

2.1 Curves and Groups of Transformations

A *curve* in $\Re^n$ is a differentiable function

$$\begin{aligned} \varphi &: I \mapsto \Re^n \\ &: \varepsilon \mapsto (\alpha_1, \dots, \alpha_n) \end{aligned}$$

where $I \subseteq \Re$ is an open interval and $\alpha_i \in \Re$ for $i = 1, \dots, n$. A curve in $V(f)$ is a curve in $\Re^n$ whose image lies in $V(f)$.

If $(\varphi^1, \dots, \varphi^n)$ is a vector field on $\Re^n$ (so $\varphi^i = \varphi^i(\vec{z})$) then for each fixed ε

$$\varphi_\varepsilon \equiv (\varphi_\varepsilon^1, \dots, \varphi_\varepsilon^n) \in \underbrace{C^1(\Re^n) \times \ldots \times C^1(\Re^n)}_{n \text{ factors}},$$

so each φ_ε determines a transformation map of $\Re^n$ given by

$$\begin{aligned} \varphi_\varepsilon &: \Re^n \mapsto \Re^n \\ &: \vec{z} \mapsto (\varphi_\varepsilon^1(\vec{z}), \dots, \varphi_\varepsilon^n(\vec{z})). \end{aligned}$$

As ε varies over I this determines a family of transformations $\{\varphi_\varepsilon\}_{\varepsilon \in I}$.

If we define the *evaluation function* at $\vec{z}$ as

$$\begin{aligned} e_{\vec{z}} &: \underbrace{C^1(\Re^n) \times \ldots \times C^1(\Re^n)}_{n \text{ factors}} \mapsto \Re^n \\ &: (f_1, \dots, f_n) \mapsto (f_1(\vec{z}), \dots, f_n(\vec{z})) \end{aligned}$$

then for a fixed ε,

$$\varphi_\varepsilon(\vec{z}) \equiv e_{\vec{z}}(\varphi_\varepsilon) = (\varphi_\varepsilon^1(\vec{z}), \dots, \varphi_\varepsilon^n(\vec{z})) \in \Re^n$$

As ε varies over I this determines a curve by

$$\begin{aligned} \varphi_\bullet(\vec{z}) \equiv e_{\vec{z}}(\varphi_\bullet) &: I \mapsto \Re^n \\ &: t \mapsto (\varphi_t^1(\vec{z}), \dots, \varphi_t^n(\vec{z})). \end{aligned}$$

In this definition $\vec{z}$ is treated as a fixed constant.

As $\vec{z}$ varies over $\Re^n$, $\varphi_\bullet(\vec{z})$ determines a family of curves, $\{\varphi_\bullet(\vec{z})\}_{\vec{z} \in \Re^n}$, one for each point $\vec{z} \in \Re^n$.

The set of transformations $\{\varphi_\varepsilon(\bullet)\}_{\varepsilon \in I}$ has a natural binary operation defined on it given by composition

$$\begin{aligned} \varphi_\varepsilon \cdot \varphi_\delta &: \Re^n \mapsto \Re^n \\ &: \vec{z} \mapsto \varphi_\varepsilon(\varphi_\delta(\vec{z})). \end{aligned}$$

A *group of transformations* $\{\varphi_\varepsilon(\bullet)\}_{\varepsilon \in I}$ is a set of transformations such that the operation of composition satisfies

i. associativity, $\varphi_\varepsilon \cdot (\varphi_\delta \cdot \varphi_\gamma) = (\varphi_\varepsilon \cdot \varphi_\delta) \cdot \varphi_\gamma$
ii. there exist an identity element φ_0, and
iii. each element in $\{\varphi_\varepsilon(\cdot)\}_{\varepsilon \in I}$ has an inverse.

The transformation $\varphi_\varepsilon(\bullet)$ is a parameterized transformation of $\Re^n$. Since it has a single parameter, the group of transformations $\{\varphi_\varepsilon(\bullet)\}_{\varepsilon \in I}$ is called a one-parameter group of transformations.

A one-parameter *Lie Group* is a group which also carries the structure of a 1-dimensional differentiable manifold. This additional structure on a group allows the ability to speak of continuity and differentiability.

2.2 Tangent Vectors and Vector Fields

A *tangent vector* consist of a vector part and a point of application. We denote a tangent vector by $\mathbf{v}_{\vec{z}} = (v_1, v_2, \dots, v_n)_{\vec{z}}$ where $(v_1, v_2, \dots, v_n)$ is "the vector part" and $\vec{z}$ is the point of application.

If φ is a curve then $\frac{d}{d\varepsilon}|_{\varepsilon=a}\varphi_\varepsilon$ determines a tangent vector at φ_a.

Each tangent vector $\mathbf{v}_{\vec{z}}$ determines a map by

$$\begin{aligned} \phi_{\mathbf{v}_{\vec{z}}} &: C^1(\Re^n) \mapsto \Re \\ &: f \mapsto \frac{d}{d\varepsilon}|_{\varepsilon=0} f(\vec{z} + \varepsilon \mathbf{v}_{\vec{z}}) \end{aligned}$$

where

$C^1(\Re^n) \equiv$ The set of differentiable functions on $\Re^n$.

For brevity we simply write

$$\mathbf{v}_{\vec{z}}(f) = \frac{d}{d\varepsilon}|_{\varepsilon=0} f(\vec{z} + \varepsilon \mathbf{v}_{\vec{z}})$$

It is an easy exercise to show that $\mathbf{v}_{\vec{z}}(f) = \frac{d}{d\varepsilon}|_{\varepsilon=0} f(\vec{z} + \varepsilon \mathbf{v}_{\vec{z}}) = \frac{d}{d\varepsilon}|_{\varepsilon=0} f(\varphi(\varepsilon))$ for any curve φ through the point $\vec{z}$ satisfying $\frac{d}{d\varepsilon}|_{\varepsilon=0}\varphi^i(\varepsilon) = \mathbf{v}^i$.

Lemma 1 *Let* $\mathbf{v} = (v_1, v_2, \dots, v_n)$ *be a vector field and* $f \in C^1(\Re^n)$. *Then*

$$\mathbf{v}(f) = \sum_{i=1}^{n} v_i \frac{\partial f}{\partial \vec{z}_i}.$$

Proof: Apply the chain rule.

By this lemma it is meaningfull to write

$$\mathbf{v}(f) = (v_1 \frac{\partial}{\partial \vec{z}_1}, v_2 \frac{\partial}{\partial \vec{z}_2}, \dots, v_n \frac{\partial}{\partial \vec{z}_n}) f$$

where

$$\mathbf{v} = v_1\frac{\partial}{\partial \vec{z}_1}, v_2\frac{\partial}{\partial \vec{z}_2}, \ldots, v_n\frac{\partial}{\partial \vec{z}_n}.$$

Thus a tangent vector, and therefore vector fields as well, can be viewed as either an ordered *n–tuple* or as an operator. It is this ability to view tangent vectors (vector fields) from both perspectives that makes them so powerful.

2.3 Killing Fields and Infinitesimal Generators

The set of vector fields over $\Re^n$ consisting of elements

$$\mathbf{v} = (v_1, v_2, \ldots, v_n)$$

where

$$v_i = v_i(\vec{z}) \in C^1(\Re^n).$$

form a module over the ring $C^1(\Re^n)$ with scalar multiplication being componentwise. Since

$$\begin{aligned} g\mathbf{v} \ &: \ C^1(\Re^n) \mapsto C^1(\Re^n) \\ &: \ f \mapsto g\mathbf{v}(f) \end{aligned}$$

where

$$\begin{aligned} (g\mathbf{v})f \ &: \ \Re^n \mapsto \Re \\ &: \ \vec{z} \mapsto (g(\vec{z})\mathbf{v}_{\vec{z}})f \end{aligned}$$

the set of all vector fields satisfying

$$\mathbf{v}(f) = 0$$

form a submodule since

$$\mathbf{v}(f) = 0 \ \& \ \mathbf{s}(f) = 0 \Rightarrow (\mathbf{v}+\mathbf{s})f = \mathbf{v}(f)+\mathbf{s}(f) = 0$$

and

$$\mathbf{v}(f) = 0 \text{ and } g \in C^1(\Re^n) \Rightarrow (g\mathbf{v})f = 0.$$

The elements of this submodule are called the *killing fields* of f. (In more standard terminology, these elements are *annihilators.* The descriptor "killing fields" is more telling of there role and will be employed here.) A collection of basis elements for this submodule are called *infinitesimal generators.*

Since the infinitesimal generators form a basis for the killing fields of f, every vector field $\mathbf{v}$ such that $\mathbf{v}(f) = 0$, with infinitesimal generators $\{\eta^1, \ldots, \eta^{n-1}\}$, can be written uniquely as

$$\mathbf{v} = \sum_{i=1}^{n-1} g^i \eta^i$$

for some $g^i \in C^1(\Re^n)$ for $i = 1, \ldots, n-1$

2.4 Computation of Groups of Transformations from the Infinitesimal Generators

Groups of transformations can be calculated from the infinitesimal generators by the following

Theorem 2 *If $\varphi_\bullet(\vec{z})$ is a curve in $V(f)$ and $\mathbf{v}$ is a vector field satisfying $\frac{d\varphi^i_\varepsilon(\vec{z})}{d\varepsilon} = v^i(\varphi_\varepsilon(\vec{z}))$ for $i = 1, \ldots, n$ then $\mathbf{v}(f) = 0$. Conversely, if $\frac{d\varphi^i_\varepsilon(\vec{z})}{d\varepsilon} = v^i(\varphi_\varepsilon(\vec{z}))$ for $i = 1, \ldots, n$, $\varphi_\bullet(\vec{z}) = \vec{z} \in V(f)$ and $\mathbf{v}(f) = 0$ then $\varphi_\bullet(\vec{z})$ is a curve in $V(f)$.*

The process of solving the equations to determine a group of transformations determined by the vector field $\mathbf{v}$ is called *the process of exponentiation.*

$$\frac{d\varphi^i_\varepsilon(\vec{z})}{d\varepsilon} = v^i(\varphi_\varepsilon(\vec{z})) \qquad \varphi^i_0(\vec{z}) = \vec{z}_i$$

for $i = 1, \ldots, n$.

Corollary 3 *Let $\mathbf{v}$ be a vector field satisfying $\mathbf{v}(f) = 0$. Each infinitesimal generator of $\mathbf{v}$ determines a curve in $V(f)$.*

Corollary 4 *Let $\{\varphi_\varepsilon\}_{\varepsilon\in\Re}$ be a group of transformations of $V(f)$ determined by the process of exponentiating. If $f(\vec{z}) = 0$ then $f(\varphi_\varepsilon(\vec{z})) = 0$.*

The conclusion of Corollary 4 is really just a tautology since a group of transformations of $V(f)$ means if $\vec{z} \in V(f)$ then $\varphi_\varepsilon(\vec{z}) \in V(f)$.

2.5 The Group of Symmetries, $S_{V(f)}$

We have observed that for an infinitesimal generator η^i of a vector field $\mathbf{v}$ satisfying $\mathbf{v}(f) = 0$, the solution to

$$\frac{d\varphi_\varepsilon(\vec{z})}{d\varepsilon} = \eta^i(\varphi_\varepsilon(\vec{z})) \qquad \varphi_0(\vec{z}) = \vec{z}$$

determines a group of transformations. If $g^i \in C^1(\Re^n)$ then $g^i\mathbf{v}^i$ is a vector field such that $g^i\mathbf{v}^i(f) = 0$ and the solution to

$$\frac{d\varphi_\varepsilon(\vec{z})}{d\varepsilon} = g^i\eta^i(\varphi_\varepsilon(\vec{z})) \qquad \varphi_0(\vec{z}) = \vec{z}$$

determines a curve in $V(f)$, and hence a group of transformations of $V(f)$. More generally, since the infinitesimal generators $\{\eta^1, \ldots, \eta^m\}$ form a basis for the vector fields $\mathbf{v}$ satisfying $\mathbf{v}(f) = 0$, then for any collection of functions

$$g^i \in C^1(\Re^n) \qquad i = 1, \ldots, n-1$$

it follows

$$\left(\sum_{i=1}^{n-1} g^i\mathbf{v}^i\right) f = 0$$

so the solution to the system of differential equations

$$\frac{d\varphi_\varepsilon(\vec{z})}{d\varepsilon} = \left(\sum_{i=1}^{n-1} g^i\eta^i\right)(\varphi_\varepsilon) \qquad \varphi_0(\vec{z}) = \vec{z}$$

determines a curve in $V(f)$, and hence a group of transformations of $V(f)$.

The set of all such transformations determined by this equation is the group of symmetries of $V(f)$, denoted by $S_{V(f)}$. Clearly it is the smallest group containing all of the groups "generated" by the infinitesimal generators $\{\eta^1, \dots, \eta^{n-1}\}$ as subgroups. Furthermore any transformation of $V(f)$ can be determined by solving such a system of equations.

2.6 Invariant Functions and their Calculation

Suppose we are given the equation $f = 0$. Let

$$\begin{aligned} \Gamma &: S_{V(f)} \times V(f) \mapsto V(f) \\ &: (\varphi_\varepsilon, \vec{z}) \mapsto \varphi_\varepsilon(\vec{z}) \end{aligned}$$

be the $S_{V(f)}$-action on $V(f)$. Then $S_{V(f)}$ acts on $hom(V(f), \Re)$ in a natural way

$$\begin{aligned} \hat{\Gamma} &: S_{V(f)} \times hom(V(f), \Re) \mapsto hom(V(f), \Re) \\ &: (\varphi_\varepsilon, \Phi) \mapsto \varphi_\varepsilon * \Phi \end{aligned}$$

where

$$\begin{aligned} \varphi_\varepsilon * \Phi &: V(f) \mapsto \Re \\ &: \vec{z} \mapsto \Phi(\varphi_\varepsilon(\vec{z})). \end{aligned}$$

Definition 5 *An element $\Phi \in hom(V(f), \Re)$ is an $S_{V(f)}$-invariant of $hom(V(f), \Re)$ if Φ is invariant under the action of $S_{V(f)}$ on $hom(V(f), \Re)$. In otherwords, the stabilizer of Φ is $S_{V(f)}$*

$$\{\varphi_\bullet \in S_{V(f)} : \varphi_\bullet * \Phi = \Phi\} = S_{V(f)}$$

It is an elementary exercise in algebra to show

Theorem 6 *Let $S_{V(f)}$ be a group acting on a set $hom(V(f), \Re)$. An element $\Phi \in hom(V(f), \Re)$ is an absolute $S_{V(f)}$-invariant of $hom(V(f), \Re)$ if and only if*

$$\Phi(\varphi_\varepsilon(\vec{z})) = \Phi(\vec{z}) \qquad \forall \varphi_\varepsilon \in S_{V(f)}.$$

Proof. See [Arnold *et al.*, 1997].

This necessary and sufficient condition is often taken as the definition of an absolute invariant function. Though the definition of an invariant element of the set $hom(V(f), \Re)$ should be expressed in terms of the more fundamental action

$$\begin{aligned} \Gamma &: S_{V(f)} \times V(f) \mapsto V(f) \\ &: (\varphi_\varepsilon, \vec{z}) \mapsto \varphi_\varepsilon(\vec{z}). \end{aligned}$$

The following theorem gives a necessary and sufficient condition for such an absolute invariant function.

Theorem 7 *Let η^i for $i = 1, \dots, n-1$ be the infinitesimal generators for the killing fields of f. Then $\Phi \in hom(V(f), \Re)$ is an absolute $S_{V(f)}$-invariant function if and only if $\eta_i(\Phi) = 0$ for $i = 1, \dots, n-1$.*

Proof. See [Arnold *et al.*, 1997].

3 Lie group analysis in Object Recognition

Several attempts at recognizing object material types using thermophysical invariance theory have been reported recently. Lie group analysis has been applied to each of the different models, including the true differential form found in previous papers [Michel *et al.*, 1997]. The following example began with the formulation presented in [Nandhakumar *et al.*, 1997], in which the radiation term was linearized and embedded into h. Further modifications (discussed below) simplified the Lie group analysis.

$$f \equiv W\alpha\cos\Theta + h(T_\infty - T_s) + K\frac{T_s - T_{int}}{\Delta y} = 0. \quad (3)$$

This model does not contain the energy storage term present in the previous models. Removal of this term allows the conservation statement to become a conservation of heat flux statement as opposed to the conservation of energy statement used before. A key reason for this fundamental shift is to find a model where the terms are independent.

A thorough analysis of the invariants of equation (3) requires the application of Lie group analysis. Consider the conservation equation (3) modeled algebraically by

$$y_1 + y_2\; y_3 - y_2\; a_1 + a_2\;\frac{a_1 - y_4}{y_5} = 0 \qquad (4)$$

where

a_1	$\equiv$	T_s	Surface temperature
a_2	$\equiv$	k	Thermal conductivity
y_1	$\equiv$	$W\ \alpha\ \cos\Theta$	Solar absorption
y_2	$\equiv$	h	Heat transfer coefficient
y_3	$\equiv$	T_∞	Ambient temperature
y_4	$\equiv$	T_{int}	Internal temperature
y_5	$\equiv$	Δy	Depth into the material (along the path of conduction)

The a_i variables are measureable (or guessed) in a recognition scenario and the y_i variables are not. Ideally we would like to find a function of the a_i variables which is an invariant.

In general, W can not be measured, while $\alpha\cos\Theta$ can be estimated. However, for the experiment discussed in the next section the entire term, $W\alpha\cos\Theta$, is measured. Also, a_2 and y_5 are constant, therefore we will have 4 transformation groups after using the equations presented in Section 2.

Generator	Transformation
v_1	$y_1 \mapsto y_1 + \varepsilon$ $a_1 \mapsto a_1 + \frac{y_5}{y_2 y_5 - a_2}\varepsilon$
v_2	$y_2 \mapsto (y_2 - \frac{a_2}{y_5})e^{\varepsilon} + \frac{a_2}{y_5}$ $a_1 \mapsto (a_1 - y_3)e^{-\varepsilon} + y_3$
v_3	$y_3 \mapsto y_3 + \varepsilon$ $a_1 \mapsto a_1 + \frac{y_2 y_5}{y_2 y_5 - a_2}\varepsilon$
v_4	$y_4 \mapsto y_4 + \varepsilon$ $a_1 \mapsto a_1 + \frac{-a_2}{y_2 y_5 - a_2}\varepsilon$

Table 1: Infinitesimal Generators and the corresponding Groups of Transformations. Note: the variables not listed under a Transformation Group undergo the identity transformation. All these transformations are global Lie groups.

The only function invariant under all the transformation groups is

$$\Phi = g[y_1 + y_2\, y_3 - y_2\, a_1 + a_2\, \frac{a_1 - y_4}{y_5}] \qquad (5)$$

where g is an arbitrary function. Hence, analytically, there are no non-trivial invariant functions for (3)! It remains to be determined if additional constraints can be found empirically such that useful quasi-invariants can be found.

4 Experimental Validation of the Group of Transformations

To check the groups of transformations found in the above application, experimental data from a thermocouple data collection performed at Wright-Patterson Air Force Base was used to determine the transformation from one data point A, to another data point B. The "ground truth" data consisted of temperature measurements acquired from thermocouples implanted in various types of materials placed in an outdoor scene and collected over a period of 2 weeks in mid-November. The collection includes varying weather conditions and has extensive records of the atmospheric pressure, ambient temperature, lighting conditions, etc. Multiple temperature measurements of sod, clay, gravel, concrete, asphalt, and aluminum were recorded every 15 minutes and provide rough estimates of all the variables in the conservation equation.

Currently, we measure and estimate all the parameters except h. Although we could also estimate h, we currently derive it from the other estimates and the conservation statement. We plan on estimating h in the future, but for this example, we found it was more useful to derive h for two reasons

1. We can check for reasonable bounds on h to verify when our model is working correctly.
2. By forcing the conservation equation to be true at each time, the transformation groups are better illustrated.

Once we have formed data points for each material at various instances in time, we can verify that our transformation groups work by solving for each ε and applying it to the surface temperature (using the appropriate transformation). If the transformations form a group (as they should), the conservation equation will hold before and after each step. By applying each of the four transformations, we can move between any two points in the group.

The missing parts of figure 1 correspond to times when the physics-based model was determined to break down. We removed these points for now since the model is not yet robust enough to consider all the different methods of heat transfer. As the model is improved, we will be able to show results for all times and include other factors such as rain, shadows, and transpiration. Ideally an extended time period of data will be used for classification since the material characteristics may be masked at any point by transient or induced effects. Only after collecting an extended period of data could one feel confident in a determination of the materials being viewed.

As previously discussed, we forced the conservation statement to hold by solving for h at each point. If h is estimated, then the resulting conservation statement will not be exactly zero, say $f(\vec{z}) = \delta$. The elements of the group of symmetries would then satisfy $f(\varphi(\vec{z})) = \delta$. A classifier would be designed to determine the threshold for which a point is considered in the class or outside the class. This is simlar to the hypothesize and verify scheme suggested in previous papers [Nandhakumar *et al.*, 1997]. However, since we can not measure all these parameters, and since we have shown non-trivial invariants do not exist, we need to look for new formulations of the model and/or quasi-invariants.

5 Discussion

5.1 New physics-based models

Another area of research is the model of the conservation equation. The current model was derived to characterize "typical" data, with no claim

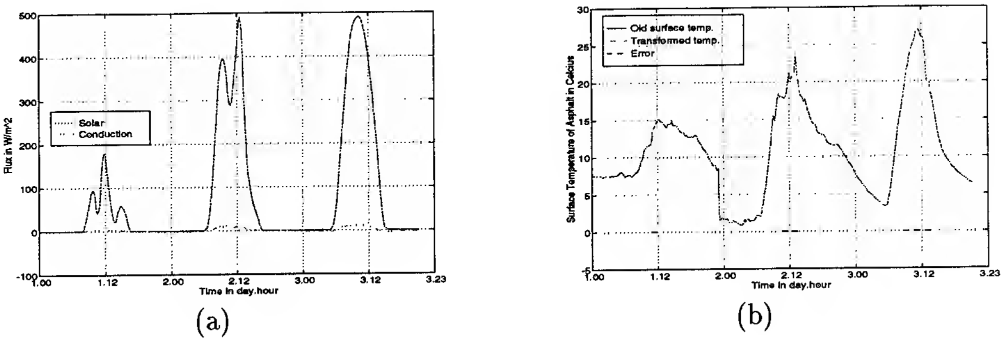


Figure 1: (a) 3 days of the solar radiation and conduction terms are shown. During the day, solar heating is clearly a dominant effect. (b) The surface temperature of asphalt before and after a 2 hour transformation is shown. The 2nd curve is shifted back 2 hours to show the exact correspondence with the original temperature, thus validating the Lie group analysis.

that it is totally accurate or complete. The model needs extensive revision and validation in order to accomplish 2 major goals

1. to include all common materials in any state (day/night, rain/shine, etc.)
2. to find a model which is both accurate and for which non-trivial invariants exist.

Since the current model clearly does not fully characterize all of the data all of the time, this will be our next step. However, it is likely that model manipulations will not reveal the absolute invariants we desire. Therefore, we must also continue research into ways of finding quasi-invariants.

5.2 Quasi-invariants

From section 2.5 it was determined that any curve in $V(f)$ must satisfy the differential equation

$$\frac{d\varphi_\varepsilon(\vec{z})}{d\varepsilon} = \left(\sum_{i=1}^{n-1} g_i\nu_i\right)(\varphi_\varepsilon) \qquad \varphi_0(\vec{z}) = \vec{z} \tag{6}$$

By curve fitting experimental data the vector fields $\frac{d\varphi_\varepsilon(\vec{z})}{d\varepsilon}$ can be determined. Since the vector fields ν_i for $i = 1, \ldots, n-1$ are known analytically, the scalar coefficients $g_i \in C^1(\Re^n)$ for $i = 1, \ldots, n-1$ can be determined. If (empirically) there is an absolute invariant then at least one of the coefficients g_i would have to be zero. This would imply they lie in a subspace of the module determined by the infinitesimal generators. This could be the result of "overlooking" some physical constraint that is not accounted for by our single equation modeling the problem – the conservation of energy equation. (One known condition we are ignoring are any bounds on the variables.) Furthermore the requirement that any curve satisfy (6) can be used to determine "quasi" (slowly varying) invariants using elementary functional analysis. Locally, if for normalized infinitesimal generators, the condition

$$||g^i||_\infty < \delta \tag{7}$$

for some i is satisfied then a function $\Phi(\bullet)$ can be determined such that $||\frac{d\Phi(\varphi_\varepsilon(\vec{z}))}{d\varepsilon}|| < \delta$. These types of invariants could be just as useful in practice as an absolute invariant.

6 Summary

The techniques of Lie group analysis provide a powerful tool for determining absolute invariant functions which can serve as classifier functions for object recognition problems. We have applied this analysis to the thermophysical invariants problem and we have proven there are no (nontrivial) absolute invariant functions for this model.

References

[Arnold *et al.*, 1997] G. Arnold, K. Sturtz, and V. Velten. Lie group analysis in object recognition (extended version). Technical report, Wright Labs, 1997.

[Michel *et al.*, 1997] J. Michel, N. Nandhakumar, and V. Velten. Thermophysical algebraic invariants from infrared imagery for object recognition. *IEEE Transactions on PAMI*, 19(1):41–51, January 1997.

[Nandhakumar *et al.*, 1997] N. Nandhakumar, Greg Arnold, Jonathan Michel, G.Tsihrintzis, and V. Velten. Robust thermophysics-based interpretation of radiometrically uncalibrated ir images for atr and site change detection. *IEEE Transactions on Image Processing*, January 1997.

[Olver, 1993] P.J. Olver. *Applications of Lie Groups to Differential Equations.* Springer-Verlag, New York, 1993.

[Sturtz, 1996] K. Sturtz. Seminar notes on algebraic invariant theory. Technical report, Wright Labs, 1996.

Monte Carlo Comparison of Distance Transform Based Matching Measures *

Daniel P. Huttenlocher
Department of Computer Science
Cornell University
Ithaca, NY 14853
dph@cs.cornell.edu

Abstract

This paper compares several measures for matching binary images that are based on distance transforms. In such measures, the "on" pixels of one binary image are used as *probes* to select distance transform values of the other image. We compare several of these measures using Monte Carlo techniques, in order to evaluate their effectiveness for matching edge images. The results demonstrate that the most effective measures use some form of outlier rejection to discard large probe values. For high probability of detection levels, the most effective of these techniques is a Hausdorff-based measure which uses a quantile of the probed distance values.

1 Introduction

This paper compares the matching performance of a number of measures for comparing binary images based on distance transforms. A distance transform of a binary image defines for each image pixel the distance to the nearest "on" pixel of that image, using a given distance function such as Euclidean distance (the L_2 norm). In order to match two images, the "on" pixels of one image are used as *probes* to select distance transform values of the other image. Thus matching measures based on distance transforms are asymmetric, because one image is used to select transform values in the other. A number of different measures are used to combine the selected distance transform values in order to determine the degree of difference (or resemblance) between two binary images. The particular measures that we consider here are defined in the following section. Such measures have formed the basis of a number of model-based recognition techiques (e.g., [3, 6]), where they are used to compare binary attributes extracted from image data. These methods have been employed in ATR systems, including for the indexing module being developed by SAIC under the MSTAR program.

We are interested in characterizing the manner in which matching measures based on distance transforms differ from one another, in terms of their ability to correctly detect a distorted instance of a target in clutter. In order to determine the power of different measures, we use Monte Carlo techniques to estimate Receiver Operating Characteristic (ROC) curves for each measure. These curves give the tradeoff between probability of detection and probability of a false alarm for the different measures, thus enabling a determinination of which measures perform better under which operating conditions. We consider variations in the amount of occlusion of the target, the amount of background clutter, the type of background clutter (correlated "edge chains" and uncorrelated points), and the spatial perturbation of the target feature points.

Supported by DARPA under Airforce contract 95040-6386 to Alphatech, and contract DAAH04-93-C-0052 to Hughes Aircraft.

2 Matching Measures

We examine the following four measures of the quality of a match between a binary model and image: (i) chamfer measure, (ii) chamfer measure with truncated distances, (iii) trimmed mean of distances, and (iv) generalized Hausdorff measure. These measure can all be defined in terms of the distance transform, $\text{dist}_I(p)$, of the binary image, I. This distance transform can be thought of as a function, parameterized by I, that yields the distance of any point from the closest feature point in I. In the current implementation we measure this distance transform using the L_1 norm (city-block distance), although other distance functions could be used. Unless otherwise noted, all of the distances in this paper will be measured with respect to this norm and the units of all distances will be image pixels. If we limit ourselves to the points on a discrete grid (such as image pixels), the distance transform of an image can be computed efficiently using a two-pass algorithm [7, 2, 6].

Let M be the set of "on" pixels of a binary model at some location in the image coordinate system. That is, in the following definitions we we assume that the pixels in the model have already been transformed (translated, rotated and scaled) to the position that we wish to compare against in the image. In addition, the coordinates of each pixel are rounded to the closest integral value.

Chamfer measure: The chamfer measure is often given as the sum of the distances from each pixel in the model to their closest image edge pixels [1]. We instead use the mean of the distances. This variation yields differences only when results are compared between different object models,

$$\text{chamf}(M,I) = \frac{1}{|M|}\sum_{m\in M} \text{dist}_I(m)\ .$$

Chamfer measure with truncated distances: It is possible to reduce the effect of occlusion by modifying the chamfer measure to be robust to outliers, by truncating the distance transform such that none of the individual distances is allowed to surpass some maximum value, d_{trunc},

$$\text{trunc}_d(M,I) = \frac{1}{|M|}\sum_{m\in M} \min(\text{dist}_I(m), d_{\text{trunc}})\ .$$

Trimmed mean of distances: Another method of making the chamfer measure robust to occlusion is to trim the largest distances before summing them. Let d_i be the i-th smallest of the probed distance transform values $D = \{\text{dist}_I(m)|m \in M\}$, that is $(d_1, \ldots, d_M)$ are the elements of D sorted into nondecreasing order, and let f be the desired fraction of values that are summed,

$$\text{trim}_f(M,I) = \frac{1}{\lfloor f\,|M|\rfloor}\sum_{i=1}^{\lfloor f|M|\rfloor} d_i\ .$$

Generalized Hausdorff measure: Rather than summing the distances, the generalized Hausdorff measure selects the fth quantile value of the set of distances [3],

$$\text{haus}_f(M,I) = \underset{m\in M}{f^{\text{th}}}\ \text{dist}_I(m)\ .$$

For example, if $f = 0.5$, the generalized Hausdorff measure selects the median of the distances.

3 Estimating ROC Curves

In order to compute an ROC curve for a given measure, we are interested in estimating the probability of detection and the probability of a false alarm over the range of possible settings of some parameter of that measure. For the chamfer measure, the only parameter is the threshold that is used to decide whether or not there is a match. The other three measures, however, each have an additional parameter besides this threshold:

- For the truncated chamfer measure, trunc_d, the value d at which the distance transform is truncated will in practice attain only a small number of values. This is because as d is increased the measure quicky approximates the regular chamfer measure. Empirical results have indicated that when there is occlusion this measure performs best when d is quite small. We thus use the fixed value $d = 2$ for the experiments.

- For the trimmed chamfer measure, trim_f, it makes sense to vary both the fraction f and the trimmed mean distance. We thus generate ROC curves by varying each of the two parameters independently. In our experiments, we held the fraction of pixels at 0.8 when the mean distance threshold was varied and held the mean distance at 0.5 pixels when the fraction of pixels threshold was varied.

- For the generalized Hausdorff measure, haus_f, the distance quantile can in practice attain only a small number of values. This is because the fraction f essentially separates inliers from outliers, and the inliers all have a small distance. Empirical results indicate that this measure performs best when the distance threshold is 1, and thus we use this fixed value.

We have estimated ROC curves for each of the measures described above by performing matching in synthetic images and using the matches found in these images to estimate the probability of detection and false alarm over the range of possible parameter settings. 1000 test images were used in the experiments, and were generated according to the following procedure. Random chains of edge pixels with a uniform distribution of lengths between 20 and 60 pixels were generated in a 256 × 256 image until a predetermined fraction of the image was covered with such chains. Curved chains were generated by changing the orientation of the chain at each pixel by a value selected from a uniform distribution between $\frac{\pi}{8}$ and $\frac{-\pi}{8}$. The chains were allowed to wrap around the edges of the image. Once the random background was so generated, an instance of a model image was placed in the image, after rotating, scaling, and translating the model image by random values. The scale change was limited to ±10% and the rotation change was limited to $\pm\frac{\pi}{18}$. Occlusion was simulated by erasing the pixels corresponding to a connected chain of the model image pixels. Gaussian noise was added to the locations of the model image pixels ($\sigma = 0.25$, unless otherwise noted). The pixel coordinates were finally rounded to the closest integer.

For the experiments reported here, we performed recognition using the 54 × 54 model image shown in Figure 1. An example of a synthetic image generated using this model image and the procedure described above is shown in Figure 2. In each trial, a given matching measure with a given parameter value was used to find all the matches of the model to the image. A trial was said to find the correct object if the position (considering only translation here) of one of the matches was within three pixels of the correct location of the object in the image. A trial was said to find a false positive if any match was found outside of this range (and that match was not contiguous with a correct match position).

3.1 Summary of Results

Overall the experiments reveal that chamfer measure works the least well of the measures, especially when there is any occlusion of the target instance. This is because the mean (or sum) is quite sensitive to a small number of moderately large distance values as occurs with partial occlusion. In the language of robust statistics, the *breakdown point* of the measure is zero, meaning that one arbitrarily large value can make the entire measure be arbitrarily large. This is a bad property when there are outliers, as occurs with partial occlusion. The other measures are all robust in the sense that they have a nonzero breakdown point, thereby enabling some number of arbitrarily large distance values to be ignored. The second overall result is that the generalized Hausdorff measure performs better than the other measures when Pd (the probability of detection) is high. Under most conditions the operating range of interest is where the Pd is high, making the generalized Hausdorff measure the most appropriate one. There is one major exception to this case, which

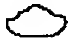


Figure 1: The model image used in the experiments.

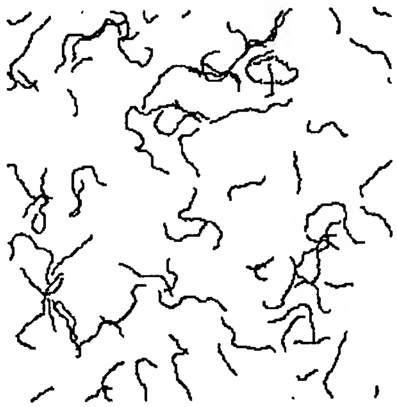
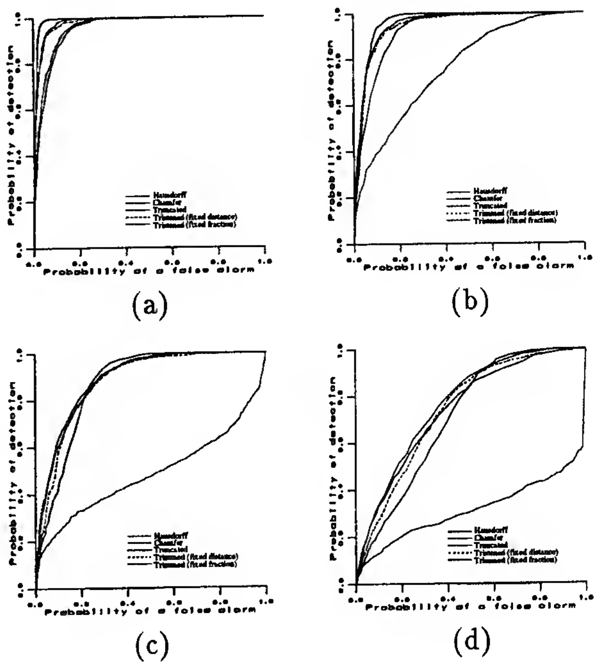


Figure 2: An example of a synthetic image that was generated with curved chains of pixel clutter. This example contains 5% clutter.

is for dense, uniformly random, point noise. In this case the generalized Hausdorff measure performs the worst of all the measures. However, uniform random noise does not generally occur in practice. In real images, noise is generally correlated.

3.2 Results

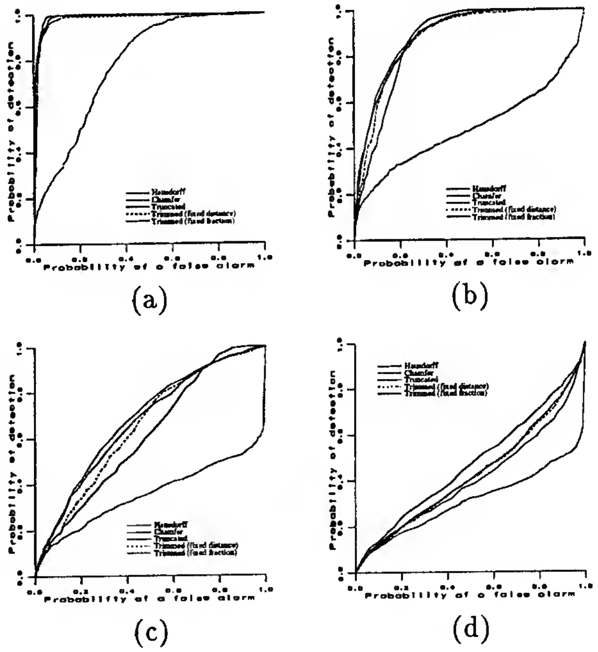
Figure 3 shows ROC curves for each of the different measures, illustrating how the performance of the measures changes with different amounts of occlusion of the object model. For each of these curves, 5% image clutter was used and the occlusion was varied from 10% through 25%. Levels of occlusion below 10% are not shown for this level of clutter, since all of the measures perform well for this case. For 10% occlusion, the generalized Hausdorff measure performs the best, with the chamfer measure and the trimmed mean measure (with a fixed fraction of 0.8) performing the worst. As the amount of occlusion is increased, the most noticeable difference is the very poor performance of the chamfer measure, which is the only one



(a) (b)

(c) (d)

Figure 3: ROC curves as a function of occlusion (fixed clutter 5%): (a) 10% occlusion, (b) 15% occlusion, (c) 20% occlusion, (d) 25% occlusion.



(a) (b)

(c) (d)

Figure 4: ROC curves as a function of clutter (fixed occlusion 20%): (a) 2.5% clutter, (b) 5% clutter, (c) 10% clutter, (d) 15% clutter.

of the measures that does not have some means of dealing with occlusion. Another notable difference is that for at least 20% occlusion, the generalized Hausdorff measure performs worse than any measure except for the chamfer measure when the the probability of detection is low, but it surpasses all of the measures when the probability of detection is high. Since recognition methods are usually concerned with the case where the probability of detection is high, these experiments indicate that the generalized Hausdorff measure will typically have the best performance.

ROC curves illustrating how the performance of the measures varies with differing levels of clutter yield similar results (not shown here). In these experiments, 20% occlusion was used, while the level of clutter was varied for 2.5% to 10%. The generalized Hausdorff measure performs best when the probability of detection is high and the chamfer measure with truncated distances performs best when the probability of detection is not high. Once again, the chamfer measure performs quite poorly when truncated distances are not used.

An additional experiment tested how the performance of the measures changed when the noise added to the object model pixels was increased. Figure 5(b) shows curves generated for the case with 5% clutter and 20% occlusion, but with increased Gaussian noise ($\sigma = 0.5$) in the localization of the object model pixels. The Hausdorff measure and the chamfer measure (which is already quite poor) are affected the least by this increase in the noise level as can be seen by comparison with Figure 5(a) which shows the $\sigma = 2.5$ perturbation that was used in the previous experiments.

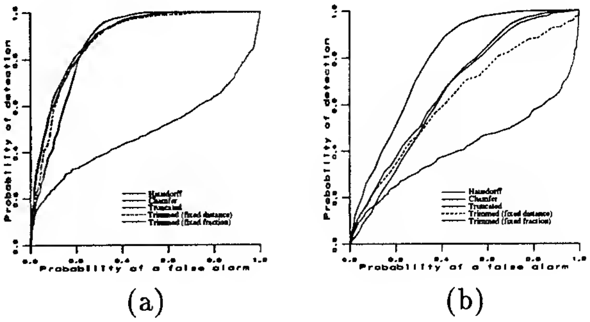


Figure 5: ROC curves for varying perturbation of the point locations with fixed clutter of 5% random curved edge segments and occlusion of 20%. (a) Gaussian noise with $\sigma = 0.25$. (b) Gaussian noise with $\sigma = 0.5$.

References

[1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.

[2] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[3] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[4] C. F. Olson and D. P. Huttenlocher. Determining the probability of a false positive when matching chains of oriented pixels. In *Proceedings of the ARPA Image Understanding Workshop*, pages 1175–1180, 1996.

[5] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, to appear 1997.

[6] D. W. Paglieroni. Distance transforms: Properties and machine vision applications. *CVGIP: Graphical Models and Image Processing*, 54(1):56–74, January 1992.

[7] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 13:471–494, 1966.

Photometric Computation of the Sign of Gaussian Curvature Using a Curve-Orientation Invariant

Elli Angelopoulou*[*] and Lawrence B. Wolff[*]

Computer Vision Laboratory, Department of Computer Science
The Johns Hopkins University, Baltimore MD 21218
{angelop, wolff}@cs.jhu.edu

Abstract

A purely geometric definition of Gaussian curvature is used for the extraction of the sign of Gaussian curvature from photometric data. Consider a point p on a smooth surface S and a closed curve γ on S which encloses p. The image of γ on the unit normal Gaussian sphere is a new curve β. The sign of Gaussian curvature at p is determined by the relative orientations of the closed curves γ and β. The relative orientation of two such curves is directly computed from intensity data. We employ three unknown illumination conditions to create a photometric scatter plot. This plot is in one-to-one correspondence with the subset of the unit Gaussian sphere containing the mutually illuminated surface normals. This permits direct computation of the sign of Gaussian curvature without the recovery of surface normals. Our method is albedo invariant. We assume diffuse reflectance, but the nature of diffuse reflectance can be general and unknown. Empirical results, demonstrate the performance of our technique.

1. Introduction

Surface curvature provides a unique three-dimensional, viewpoint-invariant description of local surface shape. Thus, curvature is a useful tool for scene analysis, feature extraction and object recognition (particularly if the scene contains sculpted, warped, free-form surfaces.) Estimates of local surface types, based on the signs of mean and Gaussian curvature, have been widely used for image segmentation and classification algorithms [1, 4, 5, 7, 14].

Extensive work has been done in the recovery of Gaussian curvature from range images. One technique involves fitting a local surface [1, 11, 12] on the range data in order to determine the partial derivatives necessary for the evaluation of Gaussian curvature. Another methodology recovers Gaussian and mean curvature from a collection of directional curvature estimates [4, 7]. However, experimental results have found that the resulting curvature estimates, are very sensitive to noise [5, 6, 9].

Gaussian curvature can be recovered from intensity images. Blake and Cipolla [2] extract curvature along apparent contours for arbitrary curvilinear viewer motion. Woodham [16] uses photometric stereo techniques to recover local surface orientation. He can then determine local curvature by taking the partial derivatives of the image irradiance equation. Wolff and Fan [3, 14] recover the sign of the Gaussian curvature without recovering the normal map. However, they assume Lambertian reflectance and they require some illumination planning.

Our method recovers the sign of Gaussian curvature directly from intensity data resulting from diffuse reflection. One of the key ideas is not to perform any surface fitting or recover the surface normals. No derivatives, or local matrices and their determinants are computed. We use the constructive geometric definition of Gaussian curvature. The Gauss map preserves orientation of closed curves around elliptical points, while it reverses orientation for hyperbolic points. Points of zero Gaussian curvature generate a closed curve which encloses a zero area.

Instead of examining the behavior of such curves on the Gaussian sphere, we can study the corresponding curves formed in photometric space. Although diffuse reflectance is assumed, the nature of the diffuse reflectance can be quite general and need not be known. The illumination conditions are also completely unknown. Each triplet of intensity values is a point in photometric space. The collection of intensity triplets for all the mutually illuminated points composes a photometric scatter plot. This scatter plot is in one-to-one correspondence with the subset of the Gaussian sphere con-

* This research was supported in part by ARPA grant DAAH04-94-G-0278, AFOSR grant F49620-93-1-0484 and the NSF National Young Investigator Award IRI-9357757.

taining the surface normals of the mutually illuminated points. Thus, the curve orientation test for the sign of Gaussian curvature can be performed directly in photometric space. This eliminates the need for recovering the surface normals. Although this technique relies on the observed brightness values, it is albedo invariant.

2. Sign of Gaussian Curvature

Given an orientable surface *S*, the *Gauss map* $N:S \to S^2$, can be thought of as taking the unit surface normal at each point $p \in S$ and "translating" it to the origin. Given a surface *S*, the Gauss map provides a framework for studying the surface normals on *S* and their directional changes.

The surface curvature at a point $p \in S$ can be measured by examining the behavior of the surface normals in a local neighborhood of *p* using the Gauss map. Consider a small simple closed curve γ on a surface *S* around a point $p \in S$. The curve γ contracts to approach the point *p*. Each point $q \in \gamma$ has a surface normal that is mapped on the unit sphere S^2. Consider now a particle on the surface *S* which is moving along the curve γ in a counterclockwise fashion. As this particle moves on *S* from one point of the curve γ to the next, it traverses a curve β on the Gaussian sphere S^2 (i.e. the curve β is composed by the endpoints of the unit surface normals of the points on γ). See fig. 1.

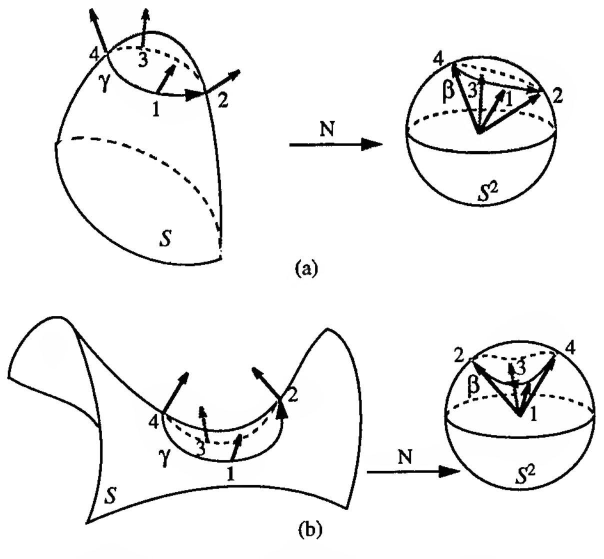


Figure 1 : (a) Preservation and (b) reversal of curve orientation on the Gauss map

Curve β can have only two possible orientations, corresponding to the two possible directions of motion along the curve. If the particle, which moves in a counterclockwise fashion on γ, traces the curve β also in a counterclockwise manner, then the Gauss map is orientation preserving at point *p*. If the particle traces the curve β in a clockwise manner, then the Gauss map is orientation reversing at point *p*. The Gaussian curvature *K* at a point $p \in S$ is positive, $K(p) > 0$, if the Gauss map is orientation preserving at *p* and negative, $K(p) < 0$, if the Gauss map is orientation reversing at *p*. For $K(p) = 0$, the area enclosed by the curve β is equal to zero.

3. Photometric Space

An image is a two-dimensional intensity pattern. A reflectance map *R* is a means of specifying the dependence of intensity values *I* on surface orientation [8]:

$$I = R(\boldsymbol{n}) \qquad (1)$$

This map combines information about surface material, scene illumination and viewing geometry into a single representation which determines the image brightness as a function of surface orientation.

Woodham [15, 16] observed in his techniques for photometric stereo that one can determine the surface normal as a function of a triplet of measured intensity values (I_1, I_2, I_3). He showed that by using a calibration object of known shape one can build a lookup table which maps measured intensity triplets to the corresponding surface normals. He explicitly recovered the mapping between the surface normals and the specific reflectance map. The recovered mapping was not albedo independent.

We never recover the map between the intensity triplets and the surface normals. We exploit only the existence of a one-to-one correspondence between triplets of photometric values and surface normals of mutually illuminated points. The specifics of the image formation process need not be known because the surface normals are not recovered. The input to the curve-orientation process is three successive images of the same scene under completely unknown illumination conditions. The only requirement is that the direction vectors of the three light sources be non-coplanar.

For each mutually illuminated pixel there is a triplet of intensity values (I_1, I_2, I_3). An intensity triplet is a point in a three dimensional *photometric space* $\Phi^3 \subseteq R^3$, where each axis represents intensity from each of the three illumination conditions. When assigning a specific axis to a unique illumination condition one should preserve the relative ordering of the light sources. The preservation of the relative ordering of the light sources guarantees that there is no basis reversal between the axes of the world coordinate system and the axes of the photometric coordinate system.

3.1 Scatter Plot

Collectively, the intensity triplets of all the mutually illuminated pixels generate a scatter plot in Φ^3. Woodham [15] showed that for constant albedo Lambertian surfaces the resulting scatter plot is a 6-degree-of-freedom ellipsoid. In general, diffuse surfaces of constant albedo exhibit gradual shading transitions: a small change in the direction of the surface normal results in a relatively small change in the observed intensity value. Consequently, the scatter plots of diffuse constant albedo objects are surfaces of positive Gaussian curvature.

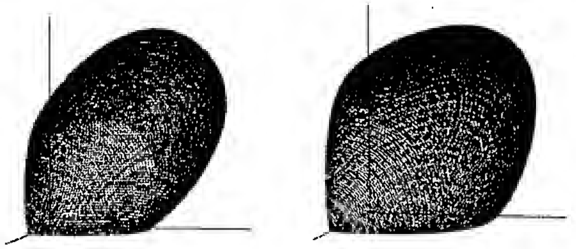

(a) (b)

Figure 2 : Photometric scatter plots for diffuse reflectance spheres of constant albedo.

For example, consider the generalizations of the Lambertian model for rough [10] and smooth [13] surfaces. Fig. 2 shows the scatter plots generated by simulating various diffuse reflectance models. Fig. 2(a) is created by a sphere with a sand-paper texture. Fig. 2(b) is created by a smooth diffuse sphere. Both plots have positive Gaussian curvature everywhere. However, when a surface exhibits specular reflectance, the shading transitions are no longer gradual. Due to specularities, small changes in the direction of the surface normal might cause abrupt changes in the observed brightness values. Thus, the scatter plots of specular reflectance objects are not necessarily surfaces of uniform sign of Gaussian curvature.

Let φ: $S^2 \rightarrow \Phi^3$ be the photometric map that transforms points on the unit Gaussian sphere S^2 to points in the photometric scatter plot Φ^3. Typically, there is a one-to-one correspondence between the intensity triplets (I_1, I_2, I_3) and the surface normals $\boldsymbol{n}$ of the mutually illuminated surface points. For constant albedo diffuse surfaces, the photometric map φ is differentiable for all points $q \in S^2$. Furthermore, for such surfaces, both the unit Gaussian sphere S^2 and the photometric scatter plot Φ^3 are surfaces of *uniformly* positive Gaussian curvature. This uniformity of the sign of Gaussian curvature in both S^2 and Φ^3 makes the photometric map φ to *uniformly* either preserve or reverse orientation for all points on S^2. The intrinsic geometry of the photometric scatter plot need not be known. The photometric map φ will be uniformly orientation reversing if and only if there is a basis reversal between the coordinate systems of the Gaussian sphere S^2 and the photometric space Φ^3. However, by construction of the photometric space, there is no such basis reversal.

The composition of the Gaussian map N and the photometric map φ creates a new map $\varphi \circ N : S \rightarrow \Phi^3$ which directly transforms mutually illuminated points on the surface S to points in the photometric space Φ^3 (see fig. 3.)

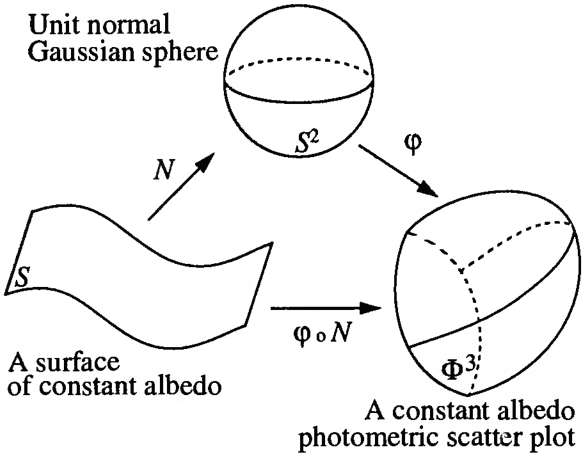


Figure 3 : A surface S and its maps on the Gaussian sphere S^2 and on the photometric space Φ^3.

Formally, by the chain rule for maps, since both the Gauss map N: $S \rightarrow S^2$ and the photometric map φ: $S^2 \rightarrow \Phi^3$ are differentiable, the composite map $\varphi \circ N : S \rightarrow \Phi^3$ is also differentiable:

$$d(\varphi \circ N)_p = d\varphi_{N(p)} \circ dN_p \tag{2}$$

For the composite map $\varphi \circ N$, the Jacobian determinant is:

$$|d(\varphi \circ N)_p| = |d\varphi_{N(p)}||dN_p| \tag{3}$$

However, the photometric map φ is orientation preserving for diffuse reflectance surfaces. Thus:

$$sign(|d(\varphi \circ N)_p|) = sign(|dN_p|) = sign(K) \tag{4}$$

This means that there is no need to recover the surface normals. We can perform the curve-orientation test on the map $\varphi \circ N$ directly.

3.2 Multiple Albedo Surfaces

So far we have ignored the effect of albedo ρ. When the same surface normal occurs at surface points which have distinct albedoes, one gets multiple triplets corresponding to the same surface normal. There is no one-to-one correspondence between the intensity triplets (I_1, I_2, I_3) and the surface normals $\boldsymbol{n}$ of the mutually illuminated surface areas. However, gnomonic projection of the scatter plot on a single plane cancels out the effects of albedo and enforces a one-to-one mapping between the Gaussian sphere S^2 and the photometric space Φ^3.

4. Curve-Orientation Test

The general idea of the curve-orientation algorithm is to construct a closed curve around each mutually illuminated pixel and test in photometric space whether its orientation is preserved or reversed, and whether it encloses an area equal to zero. We assume a right-handed image coordinate system with the z-axis pointing towards the viewer. We first create a closed-curve γ around each mutually illuminated pixel p. The curve γ is always traversed in a clockwise manner.

We then examine the corresponding curve β' that is created on the gnomonic projection plane in photometric space. The clockwise traversal of γ enforces an order in the sequence by which the list of pixels in γ is visited. We maintain this order even after the points are mapped in photometric space. Assume, for example, that pixel p_1 precedes p_2 in γ. Then $q_1 = (\varphi \circ N)(p_1)$ will be visited before $q_2 = (\varphi \circ N)(p_2)$ in β' independent of preservation of orientation. Assume that in the closed curve γ the segment (p_1, p_2) precedes (p_2, p_3). Let ν_γ be the cross-product of these segments:

$$\nu_\gamma = (p_1, p_2) \times (p_2, p_3) \qquad (5)$$

where the angle between the two segments which is facing the area enclosed by the curve is less than or equal to π. The vector ν_γ will always be pointing in the same direction with the local surface normal.

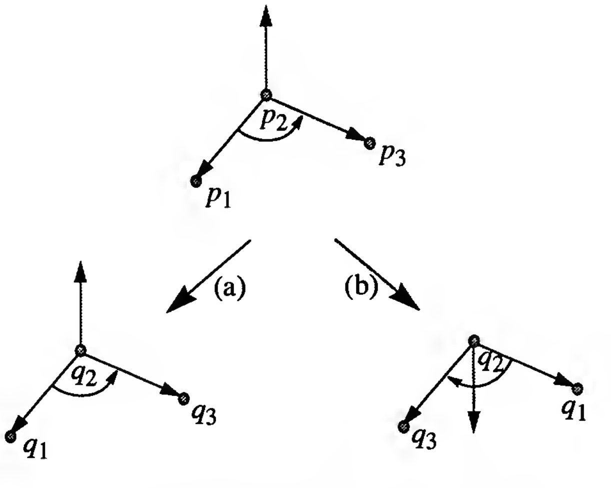


Figure 4 : (a) Preservation and (b) reversal of orientation

Accordingly, due to the enforced order, the segment (q_1, q_2) precedes the segment (q_2, q_3) on the closed curve β'. Let $\nu_{\beta'}$ be the cross-product of these segments:

$$\nu_{\beta'} = (q_1, q_2) \times (q_2, q_3) \qquad (6)$$

where the angle between the two segments which is facing the area enclosed by the curve is less than or equal to π. The vector $\nu_{\beta'}$ is always parallel to the normal of the projection plane. The map $\varphi \circ N$ preserves orientation if $\nu_{\beta'}$ points away from the origin of the photometric space and reverses orientation if $\nu_{\beta'}$ points towards that origin (see fig. 4.)

By definition in a right-handed triplet of vectors, the angle between the two vectors, which are the arguments of the cross-product operation, should be less than or equal to π. However, the closed curve β' has an arbitrary shape. The angle from a preceding line-segment to a succeeding one, which is facing the enclosed area, is not always less than or equal to π. In such a case the right-handed cross-product test can not be applied. By taking the convex hull of the points that compose the curve β' we are guaranteed that the angle between all the adjacent segments is less than π. The ordering between the segments is maintained in the resulting convex hull.

We perform this cross-product test at every point that composes the curve β'. We also examine the eccentricity of the curve β'. If the eccentricity is high, then we conclude that the curve encloses an area that approaches zero. After the sign of the Gaussian curvature is computed at every mutually illuminated pixel, a dominance averaging of the signs is performed. If there is a uniform distribution of positive, negative and zero curvature pixels in the averaging window, then it is concluded that this is a zero-curvature area. Otherwise the central pixel is assigned the sign that dominates the window.

5. Experimental Results

Our experiments indicated that our method is insensitive to the size of the window used for generating the curve on which the orientation test was performed. Various window-sizes were tried: 5×5, 7×7, and 9×9. All produced comparable results for both synthetic and real data. After the sign of the Gaussian curvature was computed at each mutually illuminated pixel, the pixels were colored according to the derived sign. Elliptic points (i.e. points of positive Gaussian curvature) are shown in medium grey, hyperbolic (i.e. points of negative Gaussian curvature) in dark grey, and parabolic and planar (i.e. points of zero Gaussian curvature) in light grey.

The experimental setup was not elaborate. The light-sources were not precision mounted. They could be placed anywhere as long as their direction vectors were not coplanar. A Sony XC-77 camera was used with a 25mm lens. Cross-polarization was used in scenes that involved objects which generated specularities. The objects were placed about 60cm from the light-sources. The curve-orientation method was tested on a variety of items made out of different materials which had either constant or multiple albedo.

One of the first objects we tested was a torus whose color had been chipped away (see fig. 5.) The torus is a good test object because it has well-defined regions of positive and negative Gaussian curvature.

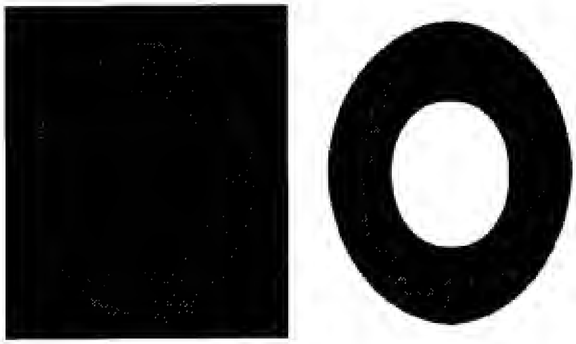


Figure 5 : An unevenly painted torus and its sign of Gaussian curvature segmentation

Among the objects we tried was a small vase with a pattern of colorful ducks and flowers. As one can see in fig. 6 the recovery of the sign of Gaussian curvature is not significantly affected by the color patterns on the surface. An uneliminated specularity at the center of the neck of the vase, causes an erroneous zero-curvature classification.

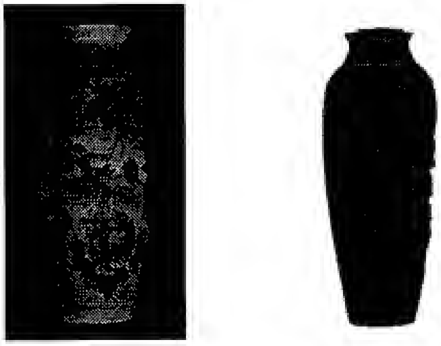


Figure 6 : A colorful vase and its sign of Gaussian curvature segmentation

Fig. 7 demonstrates the performance of our technique on a dual albedo surface which has mostly zero Gaussian curvature.

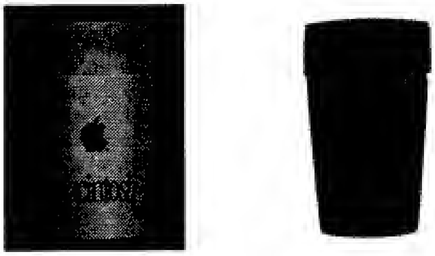


Figure 7 : A dual albedo cup and its sign of Gaussian curvature segmentation

In order to test our methodology on a surface which has multiple albedo and is not primarily a surface of zero Gaussian curvature, we constructed a free-form object out of modeling clay. We used yellow and magenta clay and created a handbag-like object. The yellow handle is a rough approximation to a segment of a torus. The main body has a concave divot in the center. Its left half is relatively flat, while its right half is more uneven. Fig. 8 shows the sign of Gaussian curvature segmentation of our clay object.

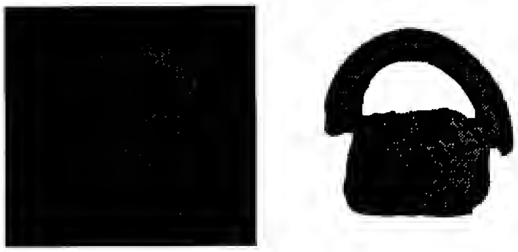


Figure 8 : A clay object and its sign of Gaussian curvature segmentation

Finally, we tried the tolerance of our algorithm in really rough surfaces. Fig. 9 shows the a small wooden carving of a bear and its segmentation. The flank is flat, but there are surface variations due to the wood grain and the carver's cuts. These are deep relative to the size of the object and are actually identified as hyperbolic areas.

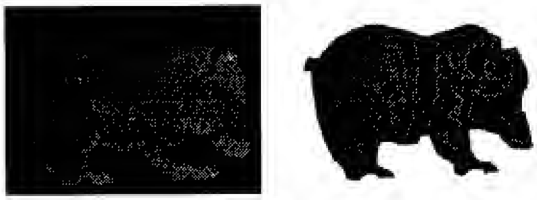


Figure 9 : A wooden carving and its sign of Gaussian curvature segmentation

6. Conclusions and Future Research

We have presented a direct method for recovering the sign of Gaussian curvature using a curve-orientation invariant. Due to its geometric nature, our technique is invariant to basis-preserving affine transformations, as well as albedo. All the computation are performed in photometric space. However, no assumptions about the model of diffuse reflectance are made. The experimental setup is inexpensive and easy to duplicate. The position of the light sources is unknown.

We are currently examining the possibility of performing our curve-orientation test directly on the photometric scatter plot. We would like to eliminate the projection to a plane, while maintaining the albedo invariance. We would also like to generalize our technique to include surfaces that exhibit specular reflectance. Finally, we are exploring the possibility of using similar direct computation techniques for the recovery of the magnitude of Gaussian curvature and the sign and magnitude of mean curvature.

References

[1] Besl, P. J. and Jain, R. C. "Invariant Surface Characteristics for 3D Object Recognition in Range Images." *Computer Vision Graphics and Image Processing,* Vol. 33, 1986. pp. 33-80.

[2] Blake, A. and Cipolla, R. "Robust Estimation of Surface Curvature from Deformation of Apparent Contours." *European Conference on Computer Vision*, 1990. pp. 465-474.

[3] Fan, J. and Wolff, L. B. "Surface Curvature from Integrability." *Computer Vision and Image Understanding*, November 1996.

[4] Fan, T., Medioni, G., and Nevatia, R. "Description of Surfaces from Range Data Using Curvature Properties." *Computer Vision and Pattern Recognition*, 1986. pp. 86-91.

[5] Flynn, P. J. and Jain, A. K. "On Reliable Curvature Estimation." *Computer Vision and Pattern Recognition*, 1989. pp. 110-116.

[6] Hilton, A., Illingworth, J. and Windeatt T. "Statistics of Surface Curvature Estimates." *International Conference on Pattern Recognition*, Vol. 1, 1994. pp. 37-41.

[7] Hoffman, R. L. and Jain, A. K. "Segmentation and Classification of Range Images." *IEEE Transactions in Pattern Analysis and Machine Intelligence,* Vol. 9, No. 5, September 1987. pp. 608-620.

[8] Horn, B. K. P. and Brooks, M. J. *Shape from Shading.* MIT Press, 1989.

[9] Lee, C. K., Haralick, R. M., and Deguchi, K. "Estimation of Curvature from Sampled Noisy Data." *Computer Vision and Pattern Recognition*, 1993. pp. 536-541.

[10] Oren, M. and Nayar, S. K. "Generalization of the Lambertian Model." *Image Understanding Workshop.* Morgan Kaufmann Publishers, 1993. pp. 1037-1048.

[11] Sander, P. T. and Zucker, S. W. "Stable Surface Estimation." *Proceedings 8th International Conference in Pattern Recognition*, October 1986. pp. 1165-1167.

[12] Vemuri, B, Mitiche, A. and Aggarwal, J. "Curvature-Based Representation of Objects from Range Data." *Image and Vision Computing*, Vol. 4, No. 2, May 1986. pp. 107-114.

[13] Wolff, L. B. "Diffuse-Reflectance Model for Smooth Dielectric Surfaces." *Journal of the Optical Society of America A*, Vol. 11, No. 11, November 1994. pp. 2956-2968.

[14] Wolff, L. B. and Fan J. "Segmentation of Surface Curvature with a Photometric Invariant." *Journal of the Optical Society of America A*, Vol. 11, No. 11, November 1994. pp. 3090-3100.

[15] Woodham, R. J. "Photometric Method for Determining Surface Orientation from Multiple Images." *Optical Engineering*, Vol. 19, No. 1, January/February 1980. pp. 139-144.

[16] Woodham, R. J. "Gradient and Curvature from the Photometric-Stereo Method, Including Local Confidence Estimation." *Journal of the Optical Society of America A*, Vol. 11, No. 11, November 1994. pp. 3050-3068.

Face *k*-D Trees for Bounded Error Point Location Operations and Surface Simplification

James P. Williams and Lawrence B. Wolff*

Computer Vision Laboratory, Department of Computer Science
The Johns Hopkins University, Baltimore MD 21218
{jimbo, wolff}@cs.jhu.edu

Abstract

Point-surface distance calculations and nearest surface point queries are essential to a broad range of geometrical applications. Registration, intersection detection in simulations and navigation all require fast solutions for these problems. It is demonstrated that *k*-dimensional (*k*-D) trees, one of the simplest and most popular structures for point location, exhibit substantial performance decreases when used on surfaces embedded in R^3. An augmented *k*-D tree, the *face k-D tree* is proposed that greatly reduces real query cost for surfaces at the expense of the introduction of a bounded amount of error in query results. A side-effect of this augmentation is also presented: the fast generation of reduced resolution surface representations suitable for visualization.

1. Introduction

Point location, the determination of the closest point or points in a given set of points to an arbitrary query location is one of the fundamental problems of Computational Geometry and an essential tool for many tasks in Computer Vision. This problem is particularly important in the registration/pose-refinement domain because algorithms like Iterated Closest Point (ICP) [1, 6, 8] rely on large numbers of nearest-neighbor queries to establish and progressively refine inter-object correspondences. Any reduction in individual query cost results in a substantial performance gain for ICP. Gains can also be realized for applications that require intersection detection of objects in R^3 such as battlefield simulation and virtual environments.

The Voronoi diagram [4], is a geometric data structure known to have a provably optimal nearest-neighbor query time of $O(logn)$. In R^2 it is always possible to construct the Voronoi diagram for a set of N points in $O(NlogN)$ time with deterministic algorithms. However, in R^3, the Voronoi diagram may have size $O(N^2)$ forcing worst case $O(N^2)$ preprocessing time. This undesirable worst-case behavior in R^3 makes Voronoi diagrams unsuitable for applications handling large point sets. For this reason and other implementation concerns, *k*-D trees, are more suitable for the point location task in R^3.

The *k*-D tree [2] has become a popular method of performing nearest-neighbor, k-nearest-neighbor and point-surface distance queries in a range of computer vision applications. The *k*-D tree is generated by recursive binary partitioning of space. At each interior node of the tree, space is partitioned by a hyperplane. The member points of the object set are found at the leaf nodes of this binary tree. This paper will deal with the specific case of R^3 and henceforth all examples can be assumed to be in R^3.

This paper will show that for the particular case of sampled surfaces embedded in R^3, the *k*-D tree displays undesirable near-worst-case behavior for queries that are not "close" to a point in the object set. The structure described in this paper, the *face k-D tree*, overcomes this drawback of conventional *k*-D trees by taking advantage of the a-priori knowledge that the object being queried is a surface. The correction comes at the expense of introducing a strictly bounded maximum potential error into the query results. A desirable side-effect is also demonstrated: if a triangulation is known for the original object surface, reduced complexity triangulations suitable for multi-resolution visualization can be produced.

2. *k*-D Trees Over Surfaces

In order to understand the shortcomings of *k*-D tree as a query structure for surface data, it is important to understand how a *k*-D tree is constructed and queried.

* This research was supported in part by ARPA grant DAAH04-94-G-0278, AFOSR grant F49620-93-1-0484 and the NSF National Young Investigator Award IRI-9357757.

The most basic form of *k*-D tree subdivides space in the following manner: First, one of the principal coordinate axes (*x, y* or *z*) is chosen as the normal direction for the splitting plane. Second, the median point, p_m, with respect to this dimension is determined. p_m can be found in linear time. The plane *P* defined by p_m and the selected coordinate axis is used to split the space into two halfspaces, each containing an equal number of points. It takes constant time to determine which halfspace a point is in, so this split can also be accomplished in linear time. The same type of subdivision is recursively applied to the two halfspaces until the number of points in the halfspace reaches some constant minimum.

In order to find the nearest neighbor to a given query point q the following algorithm is used.

```
Given a k-D tree T and a query point q

point nearest_neighbor(T,q)

  if T is a leaf node
    find the nearest neighbor n in T via brute force
    return n
  else
    get P the partition plane at T
    D := distance(P,q)
    // determine which side of P q is on
    side := which_side(P,q) // "left" or "right"
    // find the nearest neighbor on this side
    n := nearest_neigbor(T.side,q)
    if(distance(n,q) < D)
      // no need to search the other child
      return n
    else
      // must search the other child
      m := nearest_neighbor(T.~side,q)
      if(distance(q,m) < distance(q,n))
        return m
      else
        return n
```

The worst case running time of this search algorithm in R^3 is $O(N^{2/3})$ [4]. This type of behavior occurs when the distance *D* to the partition plane is frequently less than the distance to the nearest neighbor point in the object.

When the object points are clustered on a surface, query slowdown becomes evident over a large region of space. The graph in fig. 1 shows the performance of *k*-D tree nearest neighbor queries for two point sets. The first set is a uniform random distribution of points over a the cubic region *[0..100,0..100,0..100]*. The second set randomly distributes the x and y-coordinates over the same cubic region, but constrains the *z*-coordinate to lie within the range *z=[49.5..50.5]*, simulating a sampled planar surface. For each set, 10,000 queries were performed. The *x* and *y*-coordinates of the query points were randomly distributed, and the z-coordinate was evenly distributed over the range *[0..100]*. The average query cost is plotted for each value of *z*. It is clear that the query performance for the nearly planar point set approaches the performance on the random set only in a narrow band proximal to the actual *z*-range of the points.

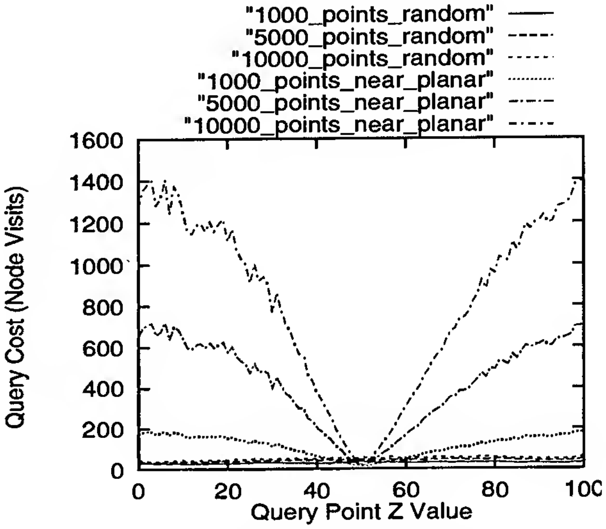


Figure 1 : Query performance with respect to position on randomly distributed and planar data.

This behavior can be explained by reasoning about the Voronoi diagram of a planar set of points embedded in R^3. The convex subdivision induced on R^3 by the Voronoi diagram can be viewed as a "perfect" k-D tree. It is never necessary to check the "outside" side of a dividing plane during a nearest neighbor search as it is in the heuristic *k*-D tree structure. Let us assume a planar point set *S* in plane *P*, and its Voronoi diagram *V* in R^3. This relationship is depicted in fig. 2.

The planar boundaries that comprise the Voronoi tessellation are projections of the lines of the *2-D* diagram in the plane parallel to the direction of the surface normal *N* of *P*. Assume that the *k*-D tree query algorithm is provided with the exact Voronoi diagram as a query structure. Even with this advantage, queries relatively distant from the plane will be closer to most of the walls of the cell than to the point enclosed by the cell, so the same pathological behavior will be manifested even given a perfect space decomposition. However, all hope is not lost for using *k*-D trees for point location on surfaces. If it is assumed that the query object is a surface, it is possible to build a special type of *k*-D tree data structure that avoids the degeneracies of conventional *k*-D trees, the face *k*-D tree.

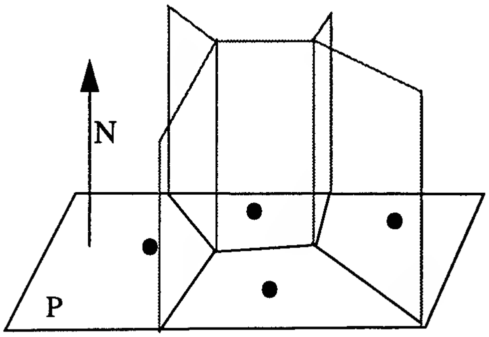


Figure 2 : A planar Voronoi Diagram and its extension into R^3

3. Error Bounded Query Structures

Suppose it is known that a query object is a surface. Surfaces embedded in R^3 have the property that in local neighborhoods, they resemble R^2, the plane. It is known that for planar point distributions, Voronoi diagrams can be constructed in provably optimal *O(NlogN)* time and queried in provably optimal *O(logN)* time [4].

Consider a set of N points in R^3 distributed within $\pm\varepsilon$ of a plane P. Parallel project these points onto P and construct the *2-D* Voronoi diagram V in P. General R^3 nearest-neighbor queries can be made by parallel projecting the query point Q onto P and performing a *2-D* query on V. This query returns an approximate nearest-neighbor. It can be shown that this approximate nearest-neighbor is less than 2ε more distant from the query point Q than the true nearest neighbor.

When dealing with the entire surface, this type of local approximate nearest-neighbor map is useful only in sub-regions of the surface that can be closely approximated by a plane. This raises the issue of how the surface should be sub-divided.

3.1 Subdividing The Object Space

The approach that is taken to subdividing the surface is similar to the technique used for creating a *k*-D tree using arbitrary partition planes. In this respect, the algorithm shares some common traits with the work of Sproull [7]. Starting with the whole unpartitioned object, the eigenvalues and eigenvectors of the covariance matrix of the object points are computed. The plane P_{min} defined by the eigenvector corresponding to the minimum eigenvalue V_{min} and the center of mass of the points is a planar approximation of the distribution of the object points in the least-squares sense. The distance from each point to P is then measured and the maximum recorded. If the maximum distance exceeds ε, then object is split by the plane P_{max} defined by the eigenvector corresponding to the maximum eigenvalue, V_{max}, and the median of the points with respect to the ordering induced by the direction of V_{max}. The subdivision continues recursively on the two equally-sized subsets which will comprise the left and right subtrees of the current node.

This subdivision process halts when all points within the current convex cell lie within epsilon of the plane P_{min} or the number of points is less than a constant threshold. If the count threshold is reached before the epsilon criterion is fulfilled, the points are placed into a "bucket" that will be searched by brute force by the query algorithm when this leaf is encountered. However, if the epsilon criterion is fulfilled first, the points are parallel-projected onto the plane P_{min} and a *2D* Voronoi diagram is constructed as the query structure for this leaf.

The cost of constructing the *2-D* Voronoi diagrams at the leaves will cost at most *O(NlogN)* and require *O(N)* space. Therefore the construction of this hybrid structure is no more costly than the construction of a conventional *k*-D tree. The following algorithm more concisely details the process for constructing a face *k*-D tree.

```
face_kd Build_Tree(num_points,points)

    if(num_points < threshold)
      return(bucket_leaf(points))
    create covariance matrix
    compute center of mass {COM}
    compute eigenvectors {$V_{min}$,$V_{mid}$,$V_{max}$}
    D = max dist. from plane ($V_{min}$,COM,points)
    if(D<$\varepsilon$)
      return(voronoi_leaf($V_{min}$,COM,points))
    else
      split points along Vmax {left,right}
      left_child =Build_Tree(num_points/2,left)
      right_child =Build_Tree(num_points/2,right)
      return interior_node(left_child,right_child)
```

3.2 Querying Face k-D Trees

The sole difference in the query algorithm for the face *k*-D tree and the conventional *k*-D query algorithm described previously occurs when a Voronoi leaf is encountered. When the query reaches such a leaf, the query point Q is parallel projected onto the plane defined by the V_{min} and center of mass of the leaf. A planar nearest-neighbor query is then performed on the Voronoi diagram V of this leaf. The point P returned by this query is know to be at most 2ε more distant from Q than any other point stored at this leaf. The result point P is referred to as an *approximate* nearest neighbor to Q.

Since the distance to the approximate nearest neighbor is always greater than or equal to the distance to the true nearest neighbor, the query algorithm will visit a superset of nodes that would have been visited had the exact nearest neighbor been found for each leaf. Therefore it can be concluded that the distance from the query point to its approximate nearest neighbor is less than 2ε greater than the distance to the true nearest neighbor.

3.3 Performance Gains

As was observed earlier, conventional k-D trees built over surfaces perform poorly when query points fall outside of a narrow range R of space that lies within some distance D_r of the surface. A face k-D tree constructed for a fixed ε will be more efficient for queries made outside of R because the size of the query structure has been reduced making the worst-case $O(N^{2/3})$ behavior less likely and less costly when it does occur. However, the hybrid structure produces inaccurate query results for points close to the surface. This is not a serious problem because although the result produced by the hybrid tree is inaccurate, it is accurate enough to determine with high probability that the query point lies within R. Since querying the conventional k-D tree is efficient within R, it is acceptable to perform exact nearest neighbor queries for those points determined to be close to the surface by less expensive approximate queries.

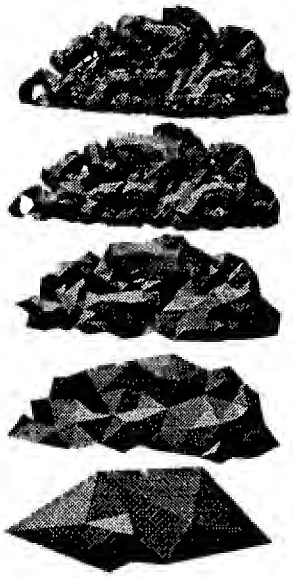


Figure 3 : Decomposition of brain set at $\varepsilon = \{0,1,2,4,8\}$

By using exact and approximate query trees in conjunction, it is possible to retain accuracy where important and to make queries fast and less accurate when a good estimate will suffice. These traits are especially important for shape registration and collision detection.

3.4 Relative Error Bounded Queries

It is desirable to maintain a statistical bound on the amount of error in the nearest neighbor calculations returned by the query structure. In this way it is possible for the user of the structure to specify the desired error relative to the distance from the query point to its nearest neighbor. It is possible to limit the percentage error by setting a threshold for the relative error E_r of approximate queries:

$$E_r = \frac{2\varepsilon}{dist(\text{query point, nearest neighbor}) - 2\varepsilon}$$

If the threshold is exceeded, the approximate query is discarded and an exact query is made. The graph in fig. 4 demonstrates the performance of a hybrid query tree on the same 5000 point planar data set used in fig. 1. The ε used in all cases was 5.0, error limits of 10%, 5% and 2% are displayed as well as the cost of using the conventional k-D tree query structure.

To further optimize query time, a series of face k-D trees can be constructed with varying values of ε. Given a series of trees $\{T_1..T_k\}$ constructed at ε values $\{\varepsilon_1..\varepsilon_k\}$ $(\varepsilon_1>\varepsilon_2>...>\varepsilon_k)$. Queries are performed by first obtaining an approximate nearest neighbor distance D from T_k. It is known that D is at most $2\varepsilon_k$ greater than the true distance. If E_r is less than the desired bound, the current point is returned. If Er is greater than the desired bound, a higher resolution tree T_i, $i>k$ is selected such that E_r will fall within the bound. The query is then repeated on tree T_i. An empirical example of the performance gain is shown in fig. 5. A series of 6 face k-D trees was constructed for a 76000 point range image of a human subject at resolutions $\varepsilon = \{5.0, 4.0, 3.0, 2.0, 1.0, 0.5\}$. A maximum query error of 2% was permitted. Query cost is plotted vs. distance from the surface for a series of 100,000 random queries made within a distance of 200 units from the surface. The head dataset is itself approximately 200 units in size so the queries fall within one object extent of the surface.

The absolute upper bound of 2ε is a pessimistic assumption. The true error of most queries will be much lower than 2ε as this bound is derived from a degenerate geometric condition that requires a high local variance of the surface that would violate most smoothness or continuity assumptions. Work is underway to tighten the bound using the signed distance D_p from the query result point to the plane onto which it was projected at a Voronoi leaf. A straightforward reduction can be made if D_p is negative (meaning that the nearest neighbor point is closer to the query point than its projection.) In this case, a reduction of the bound from 2ε to simply ε is possible.

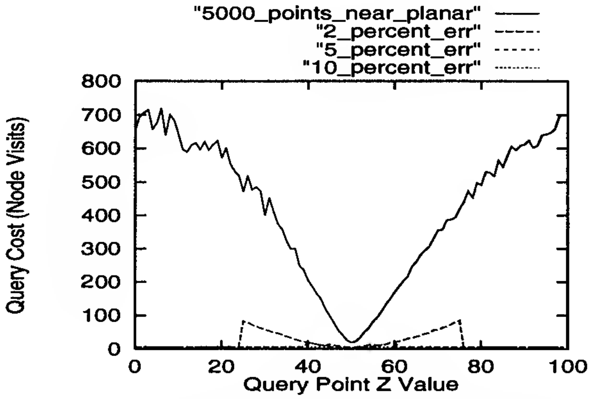


Figure 4 : Comparison of query times using face k-D Tree and conventional k-D tree on a planar distribution of points.

The two examples shown in the graphs cannot be interpreted alone as an indicator of performance improvement. However, similar speedups have been observed with empirical data in registration tasks. The performance gain is directly related to how much the surface can be simplified under a given epsilon bound. This simplification can be measured by counting the number of nodes in an approximate query tree for the given value of epsilon. Table 1 illustrates some simplification results on empirical data sets. The head set is a Cyberware scan of a human head provided by the CARD lab at Wright Patterson AFB. The lung and brain sets come from the radiology department of the Johns Hopkins University. These sets are normalized to fit within a 256x256x256 region by a uniform scaling. A bucket size of 10 is used for non-Voronoi buckets.

Table 1: Tree Size (nodes) vs. ε

ε	Head	Lung	Brain
8.0	108	66	12
4.0	279	154	62
2.0	663	464	179
1.0	1638	1170	255
0.0	7600	1622	648
# points	76000	16216	6472

The original motivation for this research was the acceleration of object registration. The face k-D representation is undergoing systematic evaluation under query loads from real registration problems. It is expected that these tests will provide a good gauge for real-world performance.

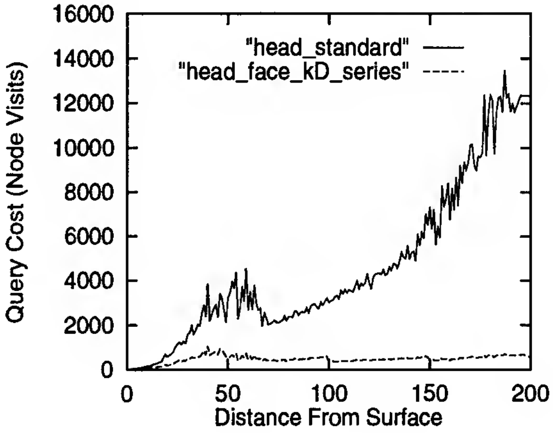


Figure 5 : Cost for uniformly distributed queries vs. distance on empirical data using standard k-D trees and a 6 tree face k-D series. Maximum allowable error for queries was 2%.

4. Variable Level of Detail Surfaces

An interesting side-effect of the hybrid k-D tree representation is that it is possible to produce surface representations at varying levels of detail directly from the tree in *O(N)* time if an initial surface triangulation is provided with the surface point set. It is the case that the majority of *3-D* point sets of surfaces are triangulated prior to use in modeling or registration applications.

Level of detail (LOD) reduction is important to real time modeling and simulation because rendering and other costs can be reduced by reducing detail for objects that are not currently a focus of attention for the system.

As a visualization tool, face k-D trees function much like another face decomposition from the graphics literature, face octrees [5]. The constructions are similar, but the rectilinear constraints imposed by the octree structure limit its ability to adapt quickly to local variation. The face k-D tree is useful for LOD reduction because it simplifies low detail flat areas of the surface into single leaves while maintaining finer subdivisions in regions

A brief description of the simplification algorithm is as follows: First, the set of centers of mass (*COMs*) of the leaf nodes is re-triangulated based on the neighbor connectivity of the original triangulation. Logically, all of the vertices within a single leaf node are merged to a single vertex located at the *COM*. If a leaf node contains points that are not a connected, planar sub-graph of the original triangulation, the leaf is split until these criterion are met. There is no assurance of topology preser-

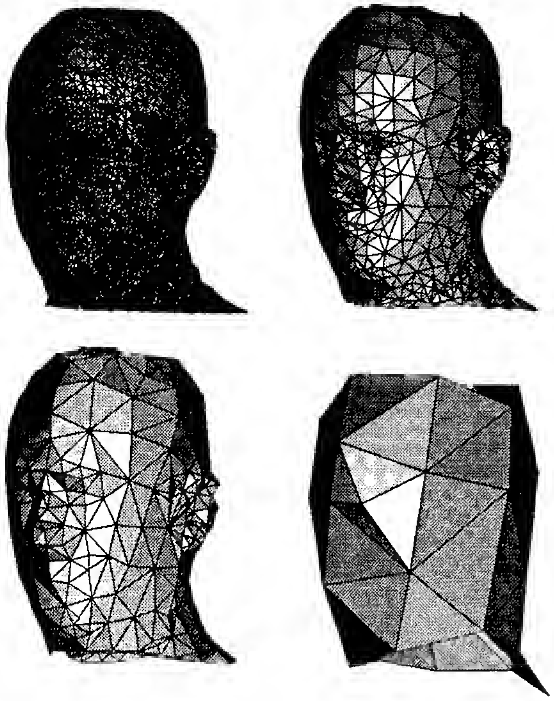


Figure 6 : Decreasing level of detail images of a human head. *(data courtesy USAF CARD Lab)*

vation in the current implementation, but work is progressing on integration of the reduced point set produced by the face *k*-D tree subdivision with existing triangulation techniques.

5. Conclusions and Future Research

This paper represents a set of preliminary results for applications of the face *k*-D tree decomposition method. It is shown how this structure provides performance superior to that of conventional *k*-D trees for point location/distance queries over surfaces embedded in R^3. For a set of N points, the cost of constructing the face *k*-D tree is $O(NlogN)$ time using $O(N)$ space. It is shown how a face *k*-D tree can be used in conjunction with a conventional *k*-D tree to allow user selection of the maximum relative error allowed in any point location query.

The research is still in its early stages and the performance increases demonstrated here are purely empirical. A formal statistical justification for the observed performance gain is forthcoming as are several geometrically motivated improvements that will improve query times.

Acknowledgments

The Cyberware scanner data of the human head was provided by the CARD lab at Wright Patterson AFB. Medical data sets come courtesy of the departments of physiology and radiology of the Johns Hopkins University Medical School. The planar Voronoi diagram implementation comes from LEDA the Library of Efficient Data types and Algorithms.

References

[1] Besl, P. and McKay, N. “A Method for the Registration of 3-D Shapes.” *IEEE PAMMI*, Vol. 14, No. 2, 1992.

[2] Bentley, J. “Multidimensional Binary Search Trees used for Associative Searching”. *Comm. ACM*, 18:509-517, 1975.

[3] Lavallee, S., Szeliski, R., and Brunie, L. “Anatomy-Based Registration of Three-Dimensional Medical Images, Range Images, X-Ray Projections and Three-Dimensional Models Using Octree-Splines.” *Computer Integrated Surgery*, pp. 114-153. MIT Press, 1996.

[4] Preperata, F. and Shamos, M. *Computational Geometry, An Introduction.* Springer-Verlag, 1986.

[5] Pla-Garcia, N. “Recovering a Smooth Boundary Representation from an Edge Quadtree and from a Face Octree.” *Computer Graphics Forum*, 13:4, pp 189-198, 1994.

[6] Simon, D., Hebert, M. and Kanade, T. “Techniques for Fast and Accurate Intrasurgical Registration.” *Journal of Image Guided Surgery*, 1:17-29, 1995.

[7] Sproull, R. “Refinements to Nearest-Neighbor Searching in k-Dimensional Trees”, *Algorithmica*, 6: 579-589, 1991.

[8] Zhang, Z. “Iterative Point Matching for Registration of Free-Form Curves and Surfaces.” *International Journal of Computer Vision*, 13:2, pp. 119-152. Kluwer, 1994.

Experiments on (Intelligent) Brute Force Methods for Appearance-Based Object Recognition

Randal C. Nelson
Andrea Selinger
Department of Computer Science
University of Rochester
Rochester, NY 14627
(nelson, selinger)@cs.rochester.edu

March 27, 1997

Abstract

It has long been recognized that, in principle object recognition problems can be solved by simple, brute force methods. However, the approach has generally been held to be completely impractical. We argue that by combining a few more or less standard tricks with computational resources that are historically large, but completely feasible by recent standards, dramatic results can be achieved for a number of recognition problems. In particular, we describe a resource-intensive, appearance-based method that utilizes intermediate-level features to provide normalized keys into a large, memorized feature database, and Bayesian evidence combination coupled with a Hough-like indexing scheme to assemble object hypotheses from the memory. This system demonstrates robust recognition of a variety of 3-D shapes, ranging from sports cars and fighter planes to snakes and lizards over full spherical or hemispherical ranges (and planar scale, translation and rotation). We report the results of various large-scale performance tests, involving, altogether, over 2000 separate test images. These include performance scaling with database size, robustness against clutter, and generic ability. The result of 97% forced choice accuracy with full orthographic invariance for 24 complex curved 3-D objects over full viewing spheres or hemispheres is the best we are aware of for this type of problem.

Key Words: Object recognition, Appearance-based representations, Visual learning.

1 Introduction

Object recognition and indexing problems have been some of the most intensely studied in the field of machine vision. Until recently, however, recognition systems, especially three dimensional ones, were quite limited in their abilities; both in the types of objects they could handle, and in the conditions under which the methods would work. It has been recognized from the beginning that, in principle, object recognition can be solved using a brute-force approach: just compare "all" possible appearances of an object directly against an image. It can even be argued that the complexity of such algorithms is linear in the number of objects. However, the constant factors in this approach are so large, (e.g. 10^{22} operations per rigid object using no smarts = 10,000 pixels x 100 intervals per DOF for 6 rigid and 3 lighting freedoms), that the approach was dismissed as completely infeasible, and work concentrated on the development of efficient algorithms. Much has been learned from this work, but what has not emerged are algorithms that are efficient enough to solve object recognition problems in any but the most limited contexts, with the sort of computer power available on, say, a 1990 desktop.

One possible conclusion is that such efficient recognition algorithms do not exist. Though there could be some undiscovered technique that will dramatically improve the situation, this seems unlikely, given the effort focussed on the problem in the last three decades. Nor is there particular evidence for the existence of such efficient algorithms in the human brain. Granted, it performs certain recognition tasks extremely well, but estimates of the computational resources it would take to simulate the relevant neural processes range from 1 to 1000+ tera-ops, with 1 to 1000 terabytes of stored information. Such estimates are extremely uncertain, but they are all large compared to a 1990 desktop, though small compared to 10^{22}.

In the field of computer science, the most dramatic change has been the phenomenal increase in the computational power of the machines which, for a fixed pricem has doubled about every 18 months for three decades. This phenomenon, sometimes referred to as Moore's Law, has repeatedly exceeded all expectations, and continues to the current date, with only a few signs of abatement.

The resulting million-fold increase has made certain algorithms, once thought impractical, practical to run. In the case of recognition, a question that has not been adequately addressed experimentally is whether near-term increases in computational resources coupled with modest algorithmic "smarts" can achieve, for certain problems, what algorithmic improvement alone has failed to do, making recognition, in one sense, a relatively "easy" problem. We think the answer to this question is yes, and furthermore that a threshold that allows interesting results to be obtained from such experiments without exclusive use of a supercomputer has recently been crossed.

In support of the position that resource intensive methods are worth looking at closely with today's power, we present some results for an appearance-based system which we believe represent the best 3-D recognition results reported anywhere for general rigid objects. The system uses only a modest amount of force (relatively speaking) and only a little of the available algorithmic cleverness, yet the results suggest not only scalable general rigid object recognition, but good performance in the presence of clutter and some surprising generic ability.

2 Background

The most successful visual recognition work to date has been using model-based systems. Notable recent examples are [11, 10, 9, 6]. The 3D geometric models on which these systems are based are both their strength and their weakness. [8, 7]. On the one hand, explicit models provide a framework that allows powerful geometric constraints to be utilized to good effect. On the other, model schemas are generally severely limited in the sort of objects that they can represent, and obtaining the models is typically a difficult and time-consuming process. There has been a fair amount of work on automatic acquisition of geometric models, mostly with range sensors, e.g., [17, 19, 2] but also visually, for various representations [20, 3, 1, 5]. However, these techniques are limited to a particular geometric schema, and even within their domain, especially with visual techniques, their performance is often unsatisfactory.

Appearance-based object recognition methods are resource-intensive algorithms that have been proposed in order to make recognition systems more general, and more easily trainable from visual data. Most of them essentially operate by comparing an image-like representation of object appearance against many prototype representations stored in a memory, and finding the closest match. They have the advantage of being fairly general, and often easily trainable. In recent work, Poggio has recognized wire objects and faces [15, 4]. Rao and Ballard [16] describe an approach based on the memorization of the responses of a set of steerable filters. Mel [12] takes a somewhat similar approach using a database of stored feature vectors representing multiple low-level cues. Murase and Nayar [13] find the major principal components of an image dataset, and use the projections of unknown images onto these as indices into a recognition memory. Schmid and Mohr [18] have recently reported good results for an appearance based system with a local-feature approach similar in spirit to what we use, though with different features and a much simpler evidence combination scheme.

In general, appearance-based methods have proven to be a useful technique; however because matches are generally made to representations of complete objects, these methods tend to be more sensitive to clutter and occlusion than is desirable, and require good global segmentation for success. Hough transform and other voting methods allow evidence from disconnected parts to be effectively combined, but the size of the voting space increases exponentially with the number of degrees of visual freedom. This makes it difficult to apply such techniques directly when more than about 3 DOF are involved, thus limiting the use of the technique for 3D object recognition, which generally involves at least 6 DOF.

We have implemented a prototype system that, by combining a large appearance database of semi-local, intermediate-level key features with a Hough-like evidence combination technique, resolves both the clutter and occlusion sensitivity of traditional memory-based methods, and the space problems of voting methods for high DOF problems. This system demonstrates robust recognition of a variety of 3-D shapes, ranging from sports cars and fighter planes to snakes and lizards over full spherical or hemispherical ranges (and planar scale, translation and rotation). It is also robust against clutter, and demonstrates some generic ability. This is in contrast to some recent results e.g. Murase and Nayar [13] where essentially only one of the two out-of-plane rotational degrees of freedom is spanned, and clutter is a significant problem.

3 The Method

3.1 Overview

The basic notion is to represent the visual appearance of an object as a structured combination of a number of semi-local features, or fragments. The idea is, that under different conditions (e.g. lighting, background, changes in orientation etc.) the feature extraction process will find some of these, but in general not all of them. However, we show that the fraction that is found by feature extraction processes is frequently sufficient to identify objects in the scene. This addresses one of the principle problems of object recognition, which is that, in any but rather artificial conditions, it has so far proved impossible to reliably segment whole objects on a bottom-up basis. In this paper, local features based on automatically extracted boundary fragments are used to represent multiple 2-D views of rigid 3-D objects, but the basic idea could be applied to other features and other representations.

In more detail, we make use of semi-invariant local objects we call *keys*. A key is any robustly extractable part or feature that has sufficient information content to specify a configuration of an associated object plus enough additional parameters to provide efficient indexing and meaningful verification. The basic idea is to utilize a database (here viewed as an associative memory) organized so that access via a key feature evokes associated hypotheses for the identity and configuration of all objects that could have produced it. These hypothesis are fed into a second stage associative memory, keyed by the configuration, which maintains a probabilistic estimate of the likelihood of each hypothesis based on statistics about the occurrence of the keys in the primary database. In our case, since 3-D objects are represented by a set of views, the configurations represent two dimensional transforms. Efficient access to the associative memories is achieved using a hashing scheme.

One step that we do not take in the current system is whole-object verification of the top hypotheses. Unlike appearance-based systems based on whole-object appearance, the structure of our representation is such that this could be performed to advantage, and such a step has the potential to significantly improve the performance of the system as a whole. The results given should thus be interpreted as representing the power of an initial hypothesis generator or indexing system.

3.2 Key Features

The recognition technique is based on the assumption that robustly extractable, semi-invariant keys can be efficiently recovered from image data. More specifically, the keys must posses the following characteristics. First, they must be complex enough not only to specify the configuration of the object, but to have parameters left over that can be used for indexing. Second, the keys must have a substantial probability of detection if the object containing them occupies the region of interest (robustness). Third, the index parameters must change relatively slowly as the object configuration changes (semi-invariance).

We currently make use of a single key feature type consisting of curve orientation templates normalized by robust boundary fragments. We call these features *curve patches*. Specifically, a curve-finding algorithm is run on an image, producing a set of segmented contour fragments broken at points of high curvature. The longest curves are selected as key curves, and a fixed-size template (21 x 21) constructed with a base segment determined by the endpoints (or the diameter in the case of closed or nearly closed curves) of the key curve occupying a canonical position in the template. All image curves that intersect the normalized template are mapped into it with a code specifying their orientation relative to the base segment. Matching of a candidate template involves taking the model patch curve points and verifying that a curve point with similar orientation lies nearby in the candidate template. Essentially this amounts to directional correlation.

3.3 Recognition Procedure

In order to recognize objects, we must first prepare a database against which the matching takes place. To do this, we first take a number of images of each object, covering the region on the viewing sphere over which the object may be encountered. The exact number of images per object may vary depending on the features used and any symmetries present, but for the patch features we use, obtaining training images about every 20 degrees is sufficient. To cover the entire sphere at this sampling requires about 100 images. For every image so obtained, the boundary extraction procedure is run, and the best 25 or so boundaries are selected as keys, from which patches are generated and stored in the database. With each patch is associated the identity of the object that produced it, the viewpoint it was taken from, and three geometric parameters specifying the 2-D size, location, and orientation of the image of the object relative to the key curve. This information permits a hypothesis about the identity, viewpoint, size, location and orientation of an object to be made from any match to the patch feature.

The basic recognition procedure consists of four steps. First, potential key features are extracted from the image using low and intermediate level visual routines. In the second step, these keys are used to access the database memory and retrieve information about what objects could have produced them, and in what relative configuration. The third step uses this information to produce hypotheses about the identity and configuration of potential objects. Finally, these hypotheses are themselves used as keys into a second associative memory, where evidence for them is accumulated. After all features have been so processed, the hypothesis with the highest evidence score is selected. Secondary hypotheses can also be reported.

3.4 Evidence Combination

In the final step described above, an important issue is the method of combining evidence. The simplest technique is to use an elementary voting scheme - each piece of evidence contributes equally to the total. This is clearly not well founded, as a feature that occurs in many different situations is not as good an indicator of the presence of an object as one that is unique to it. For example, with 24 3-D objects stored in the database, comprising over 30,000 patches, we find that some image features match 1000 or more database features, while others match only one or two. An evidence scheme that takes this into account would probably display improved performance. An obvious approach in our case is to use statistics computed over the information contained in the associative memory to evaluate the quality of a piece of information. It is clear that the optimal quality measure, which would rely on the full joint probability distribution over keys, objects and configurations is infeasible to compute, and thus we must use some approximation.

A simple example would be to use formally, the first order feature frequency distribution over the entire database, and this is what we do. The actual algorithm is to accumulate evidence, for each match supporting a pose, proportional to $F \log(k/m)$ where m is the number of matches to the image feature in the whole database, and k is a proportionality constant that attempts to make m/k represent the actual geometric probability that some image feature matches a particular patch in the pose model by accident. It can be shown that maximizing the summed reciprocal log terms is equivalent to Bayesian maximum likelihood evidence combination using the match frequency as an estimate of the prior probability of the the feature type, and assuming independence of observations. F represents an additional empirical factor proportional to the square root of the size of the feature in the image, and the 4th root of the number of key features in the model. These modifications capture certain aspects that seem important to the recognition process, but are difficult to model using formal probability, (essentially that bigger features are better, and that the simplest explanation is preferred)

The above measure allows us to combine evidence for all feature matches associated with a given pose hypothesis and a set of evidence. We now want to find the maximum of this over all possible poses. Clearly, we can't directly evaluate all pose hypotheses: there are too many of them (e.g. 20 objects x 100 viewpoints x 100 image locations x 20 orientations x 10 sizes = 40,000,000 poses to check). In our algorithm, the indexing into the secondary associative memory functions as an efficient way of accumulating the evidence for all poses that have any evidence associated with them at all (most possible poses have none, for a given set of evidence). This is the basic Hough transform idea, and it permits the pose with maximum evidence to be found in time proportional to the number of pieces of evidence times a database lookup factor rather than in time proportional to the number of possible poses.

3.5 Implementation

Using the principles described above, we implemented a recognition system for rigid 3-D objects. The system needs a particular shape or pattern to index on, and does not work well for objects whose character is statistical, such as generic trees or pine cones. Component boundaries were extracted by modifying a stick-growing method for finding segments developed recently at Rochester [14] so that it could follow curved boundaries.

The system is trained using images taken approximately every 20 degrees around the sphere, amounting to about 100 views for a full sphere, and 50 for a hemisphere. of making the templates sufficiently flexible to match between views. For objects entered into the database, the best 25 key features were selected to represent the object in each view. The thresholds on the distance metrics between features were adjusted so that they would tolerate approximately 15-20 degrees deviation in the appearance of a frontal plane (less for oblique ones).

The system can be used both for "recognition" (what is this) and "finding" (where is object X in this scene) operations. Preliminary experiments on the finding operation indicate good performance in large, complex scenes, but we have not yet acquired even moderate test databases for this problem, so the experiments reported below all involve "recognition" tasks.

/subsectionResource Requirements The resource requirements scale more or less linearly with the size of the database. Memory is about 3 Mbytes per hemisphere, and overall times on a single processor Ultrasparc are about 20 seconds for the 6 object database, and about 2 minutes for the 24 object database. These numbers could almost certainly be improved by pushing on the indexing and data replication, which we have not done as yet.

4 Experiments

4.1 Variation in Performance with Size of Database

One measure of the performance of an object recognition system is how the performance changes as the number of classes increases. To test this, we obtained test and training images for a number of objects, and built 3-D recognition databases using different numbers of objects. The objects used were chosen to be "different" in that they were easy for people to distinguish on the basis of shape. Data was acquired for 24 different objects (34 hemispheres). The objects are shown in Figure 1. The number of hemispheres is not equal to twice the number of objects because a number of the objects were either unrealistic or painted flat black on the bottom which made getting training data against a black background difficult.

Clean image data was obtained automatically using a combination of a robot-mounted camera, and a computer controlled turntable covered in black velvet. Training data consisted of 53 images per hemisphere, spread fairly uniformly, with approximately 20 degrees between neighboring views. The test data consisted of 24 images per hemisphere, positioned in between the training views, and taken under the same good conditions. Note that this is essentially a test of invariance under out-of-plane rotations, the most difficult of the 6 orthographic freedoms. The planar invariances are guaranteed by the representation, once above the level of feature extraction, and experiments testing this have shown no degradation due to translation, rotation, and scaling up to 50%. Larger changes in scale have been accommodated using a multi-resolution feature finder, which gives us 4 or 5 octaves at the cost of doubling the size of the database.

We ran tests with databases built for 6, 12, 18 and 24 objects, shown in Figure 1, and obtained overall success rates (correct classification on forced choice) of 99.6%, 98.7% 97.4% and 97.0% respectively. (To find out which objects are in which database, just count the images left to right, top to bottom.) The results are summarized in the following table. The worst cases were the horse and the wolf in the 24 object test, with 19/24 and 20/24 correct respectively. On inspection, some of these pictures were difficult for human subjects. None of the other examples had more than 2 misses out of the 24 (hemisphere) or 48 (full sphere) test cases. Overall, the performance is fairly good. In fact, we believe this represents the best results presented anywhere for this sort of problem.

num. of objects	num. of hemi-spheres	num. of test images	num. correct	percent correct
6	11	264	263	99.6
12	18	408	403	98.7
18	26	576	561	97.4
24	34	768	745	97.0

Table 1: Performance of forced-choice recognition for databases of different sizes

4.2 Performance in the Presence of Clutter

The feature-based nature of the algorithm provides some immunity to the presence of clutter in the scene, in contrast to appearance-based schemes that use the structure of the full object, and require good global segmentation. For modest dark-field clutter, the method is quite robust. To test this, we acquired test sets of the six objects used in the previous 6-object case in the presence of non-occluding clutter. Examples of the test images are shown in Figure 2 Out of 264 test cases, 252 were classified correctly which gives

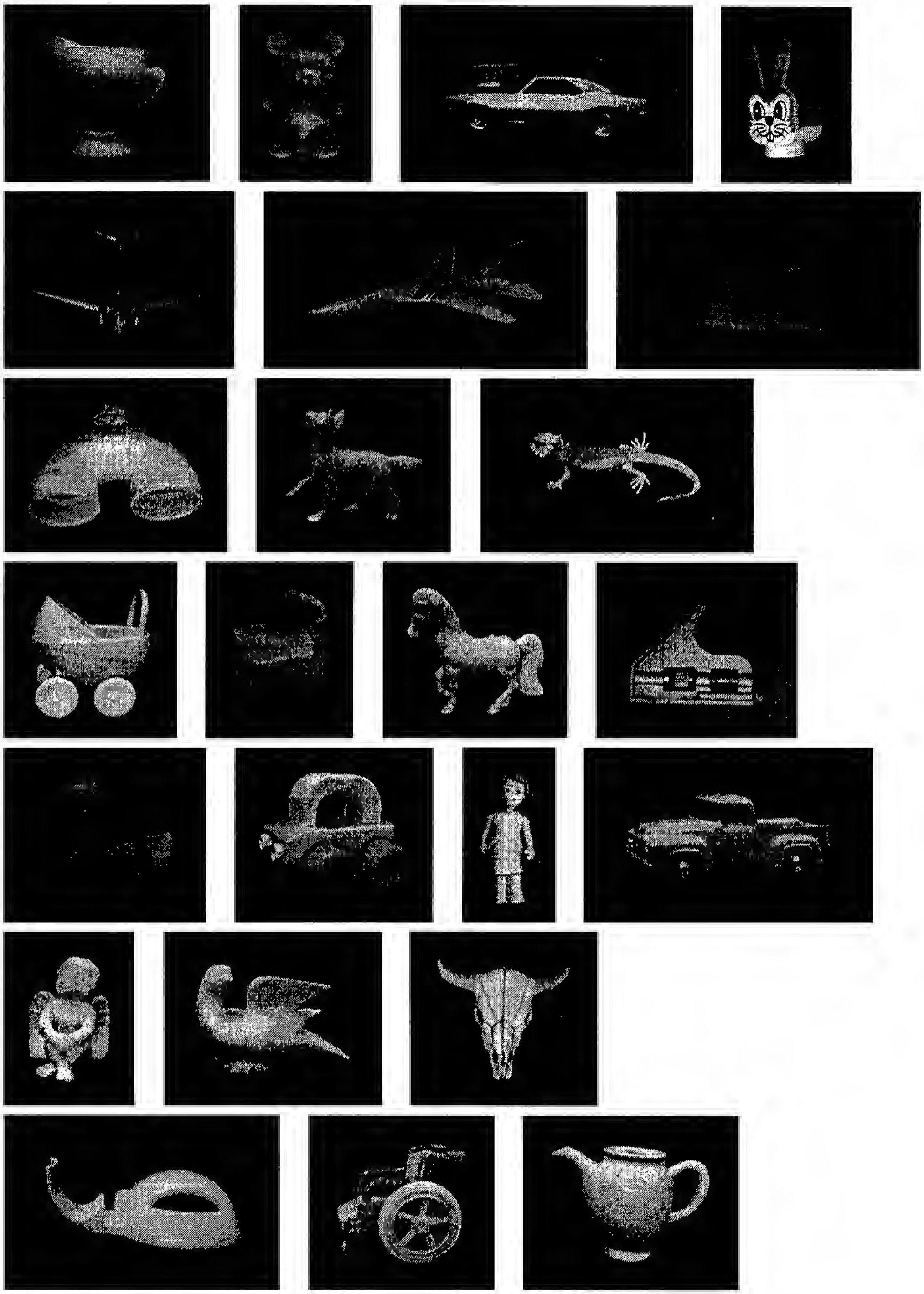


Figure 1: The objects used in testing the system

a recognition rate of about 96%, compared to 99% for uncluttered test images. A confusion matrix is shown in Figure 3

class	ref	num	0	1	2	3	4	5
cup	0	48	47	0	1	0	0	0
bear	1	48	2	46	0	0	0	0
car	2	24	0	0	24	0	0	0
rabbit	3	48	0	0	1	47	0	0
plane	4	48	0	0	2	1	45	0
fightr	5	48	0	0	1	0	4	43
Hypoths. for class			49	46	29	48	49	43

Figure 3: Error matrix for object classification experiment with clutter. Columns contain counts of classification results for test images of each type.

In a second experiment, we took pictures of the objects against a light background. Clutter in these images arises from shadows, from wrinkles in the fabric, and from a substantial shading discontinuity between the turntable and the background. Unlike the dark-field pictures, the objects in many of these pictures are not trivially segmentable. Examples of the test images are shown in Figure 4, and the boundaries found in Figure 5. Note that some of the images produce substantial numbers of clutter curves. (All the images shown were classified correctly.)

Out of 264 test cases, 236 were classified correctly which gives an overall recognition rate of about 90%, which is not as good as some of our other results. However, almost half the errors were due to instances of the toy bear, the reason being that the gray level of the bear's body was so close to the upper background in low-level shots that many of the main boundaries could not be found (people had trouble with these shots too). If this case is excluded, the rate is about 94%. A confusion matrix is shown in Figure 6

class	ref	num	0	1	2	3	4	5
cup	0	48	44	2	0	1	1	0
bear	1	48	3	32	1	5	2	5
car	2	24	0	0	24	0	0	0
rabbit	3	48	1	0	0	47	0	0
plane	4	48	0	0	0	0	45	3
fightr	5	48	0	0	1	0	3	44
Hypoths. for class			48	34	26	53	51	52

Figure 6: Error matrix for light field classification experiment. Columns contain counts of classification results for test images of each type.

4.3 Experiments on "Generic" Recognition

This set of experiments was suggested when, we tried showing our coffee mugs to an early version of the system that had been trained on the creamer cup in the previous database (among other objects), and noticed that the system was making the "correct" generic call a significant percentage of the time. Moreover, the features that were keying the classification were the "right" ones, i.e., boundaries derived from the handle, and the circular sections, even though there was no explicit part model of a cup in the system.

The notion of generic visual classes is ill defined scientifically. What we have is human subjective impressions that certain objects look alike, and belong in the same group (e.g. airplanes, sports cars, spiders, teapots etc.) Unfortunately, human visual classes tend to be confounded with functional classes, and biased by experience and other factors to an extent that makes formalizing such classes, even phenomenologically, pretty tough. On the other hand, the subjective intuition is so strong, and the early evidence of correct "generalization" so intriguing, that the matter seemed worth looking into.

For the test, we gathered multiple examples of objects from several classes, which an (informal) sample of human volunteers agreed looked pretty much alike (our rough criterion was you could tell at a glance what class an object was in, but had to take a "second look" to determine which member of the class it was. We ended up with five classes consisting of 11 cups, 6 "normal" airplanes, 6 fighter jets, 9 sports cars, and 8 snakes.

The recognition system was trained on a subset of each class, and tested on the remaining elements. The training sets consisted of 4 cups, 3 airplanes, 3 jet fighters, 4 sports cars, and 4 snakes. These classes are shown in Figure 7, with the training objects on the left of each picture, and the test objects on the right. The training and test views were taken according to the same protocol as in the previous experiment. The cups, planes, and fighter jets were sampled over the full sphere; the cars and snakes over the top hemisphere (the bottom sides were not realistically sculpted). Overall performance on forced choice classification for 792 test images was 737 correct, or 93.0%. If we average performance for each group so that the fact that the best group, the cups, does not get weighted more because we had more samples, we get 92% (91.96%) performance. The error matrix is shown in Figure 8

The performance is best for the cups at about 98%, and the planes, sports cars and snakes came in around 92%-94%. The fighter planes were the worst by a significant factor, at about 83%. The reason seems to be that there is quite a bit of difference between the exemplars in some views in terms of armament carried, which tends to break up some of the lines in a way the current boundary finder does not handle. Two of the test cases also have camouflage patterns painted on them. The snakes were actually a bit of a surprise, given the degree of flexibility, and the fact that none of the curves are actually the same (this is supposedly a rigid object recognition system). The key seems to be

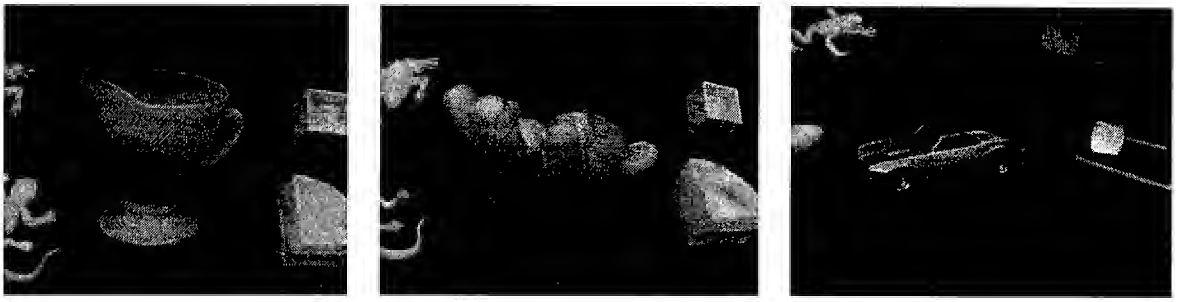


Figure 2: Examples of test images with modest dark-field clutter

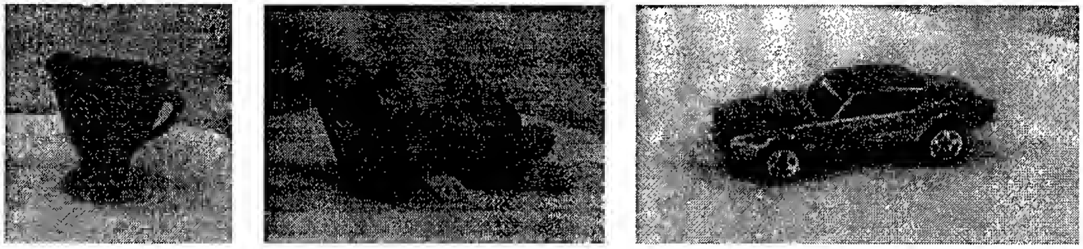


Figure 4: Examples of test images on light background, with shadows and minor texture

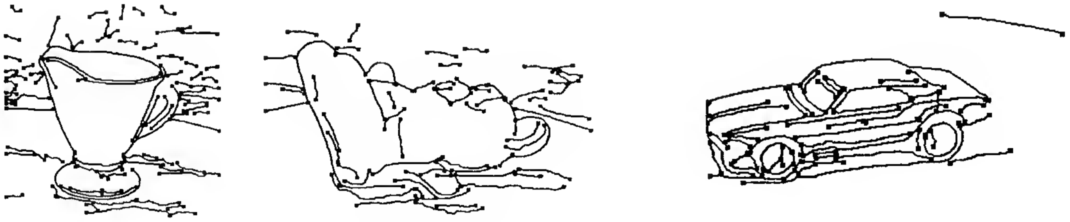


Figure 5: Curves found by boundary extraction algorithm in light background images

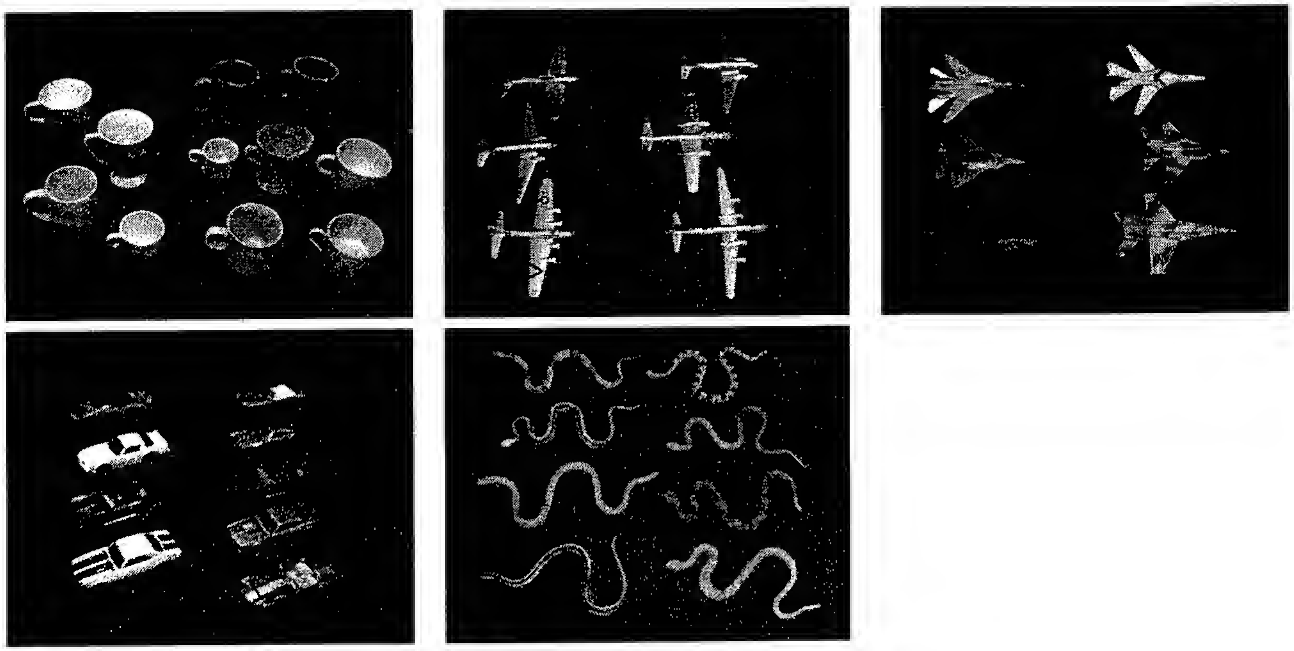


Figure 7: Test sets used in generic recognition experiment. The training objects are on the left side of each image (4 cups, 3 planes, 3 fighters, 4 cars, 4 snakes) and the test objects are on the right.

class	ref	num	0	1	2	3	4
cup	0	288	282	0	6	0	0
fightr	1	144	0	120	7	16	1
snake	2	96	5	0	88	1	2
plane	3	144	0	2	7	135	0
car	4	120	1	0	6	1	112
Hypoths. for class			288	122	114	153	115

Figure 8: Error matrix for generic classification experiment. Columns contain counts of classification results for test images of each type.

the generic "S" shape, which recurs in various ways in all the exemplars, and is quite rare in general scenes.

These results do not say anything conclusive about the nature of "generic" recognition, but they do suggest a route by which generic capability could arise in an appearance based system that was initially targeted at recognizing specific objects, but needed enough flexibility to be able to deal with inter-pose variability and environmental lighting effects. They also suggest that one way of viewing generic classes is that they correspond to clusters in a (relatively) spatially uniform metric space defined by a general, context-free, classification process. Finer distinctions would make use of this context.

5 Conclusions and Future Work

In this paper we have described a framework for keyed appearance-based 3-D recognition, which avoids some of the problems of previous appearance-based schemes. We ran various large-scale performance tests and found good performance for full-sphere/hemisphere recognition of up to 24 complex, curved objects, robustness against clutter, and some intriguing generic recognition behavior.

Future plans include adding enough additional objects to push the performance below 75%, both to better observe the functional form of the error dependence on scale, and to provide a basis for substantial improvement. We also want to see how the performance can be improved by adding a final verification stage, since we have observed that even when the system provides the wrong answer, the "right" one is generally in the top few hypotheses. In another direction, we have some preliminary results indicating that the system, when coupled with a simple memory-constraint protocol, functions very well for finding particular objects in large, highly cluttered scenes. We plan to gather enough data for this problem to generate statistically significant performance data. Finally, we want to experiment with adapting the system to allow fine discrimination of similar objects (same generic class) using directed processing driven by the generic classification.

References

[1] Nicholas Ayache and Olivier Faugeras. Hyper: a new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. PAMI*, 8(1):44–54, January 1986.

[2] Aaron F. Bobick and Robert C. Bolles. Representation space: An approach to the integration of visual information. In *Proc. CVPR*, pages 492–499, San Diego CA, June 1989.

[3] Robert C. Bolles and R. A. Cain. Recognizing and localizing partially visible objects: The local-features-focus method. *International Journal of Robotics Research*, 1(3):57–82, Fall 1982.

[4] R. Brunelli and Thomaso Poggio. Face recognition: Features versus templates. *IEEE Trans. PAMI*, 15(10):1042–1062, 1993.

[5] F.Stein and Gerard Medioni. Efficient 2-dimensional object recgnition. In *Proc. ICPR*, pages 13–17, Atlantic City NJ, June 1990.

[6] W. E. L Grimson. *Object Recognition by Computer: The role of geometric constraints.* The MIT Press, Cambridge, 1990.

[7] W. E. L. Grimson and Danial P. Huttenlocher. On the sensitivity of the hough transform for object recognition. *IEEE PAMI*, 12(3):255–274, 1990.

[8] W. E. L. Grimson and Daniel P. Huttenlocher. On the sensitivity of geometric hashing. In *3rd International Conference on Computer Vision*, pages 334-338, 1990.

[9] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[10] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. International Conference on Computer Vision*, pages 238–249, Tampa FL, December 1988.

[11] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[12] Bartlett Mel. Object classification with high-dimensional vectors. In *Proc. Telluride Workshop on Neuromorphic Engineering*, Telluride CO, July 1994.

[13] Hiroshi Murase and Shree K. Nayar. Learning and recognition of 3d objects from appearance. In *Proc. IEEE Workshop on Qualitative Vision*, pages 39–50, 1993.

[14] Randal. C. Nelson. Finding line segments by stick growing. *IEEE Trans PAMI*, 16(5):519–523, May 1994.

[15] Thomaso Poggio and Shimon Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 1990.

[16] Rajesh P.N. Rao. Top-down gaze targeting for space-variant active vision. In *Proc. ARPA Image Understanding Workshop*, pages 1049–1058, Monterey CA, November 1994.

[17] R. Kjeldsen Ruud M. Bolle and Daniel Sabbah. Primitive shape extraction from range data. In *Proc. IEEE Workshop on Computer Vision*, pages 324–326, Miami FL, Nov-Dec 1989.

[18] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proc. CVPR96*, pages 872–877, San Francisco CA, June 1996.

[19] F. Solina and Ruzena Bajcsy. Recovery of parameteric models from range images. *IEEE Trans. PAMI*, 12:131–147, February 1990.

[20] Shimon Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Trans. PAMI*, 13(10), 1991.

Recognizing Objects by Matching Oriented Points

Andrew Edie Johnson and Martial Hebert
The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213

We present an approach to recognition of complex objects in cluttered 3-D scenes that does not require feature extraction or segmentation. Our object representation comprises descriptive images associated with each oriented point on the surface of an object. Using a single point basis constructed from an oriented point, the position of other points on the surface of the object can be described by two parameters. The accumulation of these parameters for many points on the surface of the object results in an image at each oriented point. These images, localized descriptions of the global shape of the object, are invariant to rigid transformations. Through correlation of images, point correspondences between a model and scene data are established. Geometric consistency is used to cluster the correspondences from which a plausible rigid transformation that align the model with the scene is calculated. The transformations is then refined and verified using a modified iterative closest point algorithm. The effectiveness of our algorithm is demonstrated with results showing recognition of complex objects in cluttered scenes with occlusion.

1. Introduction

For recognition of complex objects, we have developed a representation that combines the descriptive nature of global object properties with the robustness to partial views and clutter of local shape descriptions. Specifically, a local basis is computed at an oriented point (3-D point with surface normal) on the surface of an object represented as a polygonal surface mesh. The positions with respect to the basis of other points on the surface of the object can then be described by two parameters. By accumulating these parameters in a 2-D array, a descriptive image associated with the point is created. Because the image describes the coordinates of points on the surface of an object with respect to the local basis, it is a local encoding of the global shape of the object and is invariant to rigid transformations. To prepare a model for recognition, an image is generated for each point on the model. Since an image is generated at each point in the surface mesh, error prone feature extraction and segmentation are avoided. At recognition time, images from points on the model are compared with images from points in the scene; when two images are similar enough, a point correspondence between model and scene is established. Several point correspondences are then used to calculate a transformation from model to scene for verification.

Our recognition technique developed from a combination of basis geometric hashing proposed by Lamdan and Wolfson [11] and structural indexing proposed by Stein and Medioni [13]. Because we use information from the entire surface of the object in our representation, instead of a curve or surface patch in the vicinity of the point, our representation is more discriminating than the curves used to date in structural indexing. Furthermore, because bases are computed from single points, our method does not have the combinatoric explosion present in basis geometric hashing as the amount of points is increased. In our algorithm, every point on the model that is visible in the scene can be matched. This is in contrast to geometric hashing where only select feature points can be matched, making its effectiveness dependent on feature extraction.

The idea of encoding the relative position of many points on the surface of an object in an image or histogram is not new. Ikeuchi et. al. [10] propose invariant histograms for SAR target recognition. This work is view-based and requires feature extraction. Gu éziec and Ayache [5] store parameters for all points along a curve in a hash table for efficient matching of 3-D curves. Their method requires the extraction of extremal curves from 3-D images.

Chua and Jarvis [3] present an algorithm for matching 3-D free-form surfaces by matching points based on principal curvatures. Similarly, Thirion [15] presents an algorithm for matching 3-D images based on the matching of extremal points using curvatures and Darboux frames. Pipitone and Adams [12] propose the tripod operator which, when placed on the surface of an object, generates a few parameters describing surface shape. Bergevin et. al. [2] propose a registration algorithm based on matching properties of triangles generated from a hierarchical tessellation of an object's surface. Our approach differs from these because the images computed at each point are much more discriminating than principal curvatures and angles between frames measured at a point. The discriminability of spin-images greatly reduces the number possible correspondences between points.

During recognition, a large number of points over the entire scene are matched to model points. Since many correspondences are established, the validity of individual correspondences can be determined by comparing them to the properties of the group of correspondences. Using this reasoning our algorithm is able to select the best correspondences even in the presence of complex scenes with clutter and occlusions.

The paper is organized as follows: First, we describe how the images used in matching oriented points are generated and compared. Next, we explain the robustness of the images to clutter and occlusion and describe how correspondences are grouped to compute plausible transformations from model to scene. We then explain our verification algorithm, which is based on the iterative closest point algorithm. Finally, we show recognition results in cluttered scenes and conclude with a discussion of future work.

2. Spin-Images

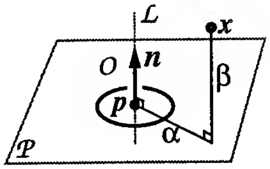


Figure 1: The geometry of an oriented point basis.

The fundamental shape element we use for matching is an **oriented point**, a three-dimensional point with an associated direction. We define an oriented point O on a surface mesh of an object using vertex position $\boldsymbol{p}$ and surface normal $\boldsymbol{n}$ (defined as the normal of the best fit plane to the point and its neighbors in the mesh oriented to the outside of the object). As shown in Figure 1, an oriented point defines a 2-D basis $(\boldsymbol{p},\boldsymbol{n})$ (i.e., local coordinate system) using the tangent plane $\mathcal{P}$ through $\boldsymbol{p}$ oriented perpendicularly to $\boldsymbol{n}$ and the line $\mathcal{L}$ through $\boldsymbol{p}$ parallel to $\boldsymbol{n}$. The two coordinates of the basis are α, the perpendicular distance to the line $\mathcal{L}$, and β the signed perpendicular distance to the plane $\mathcal{P}$. A ***spin-map*** S_O is the function that maps 3-D points $\boldsymbol{x}$ to the 2-D coordinates of a particular basis $(\boldsymbol{p},\boldsymbol{n})$ corresponding to oriented point O

$$S_O(\boldsymbol{x}) = \left(\sqrt{\|\boldsymbol{x}-\boldsymbol{p}\|^2 - (\boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{p}))^2}, \boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{p})\right) \quad (1)$$

The term spin-map comes from the cylindrical symmetry of the oriented point basis; the basis can spin about its axis with no effect on the coordinates of points with respect to the basis.

Each oriented point O on the surface of an object has a unique spin-map S_O associated with it. When S_O is applied to all of the other points on the surface of the object $\mathcal{M}$, a set of 2-D points is created. We will use the term **spin-image** $I_{O,\mathcal{M}}$ to refer to the result of applying the spin-map S_O to the set of points on $\mathcal{M}$. A spin-image is a description of the shape of an object because it is the projection of the relative position of 3-D points that lie on the surface of an object to a 2-D space where some of the 3-D metric information is preserved. Since spin-images describe the shape of an object independently of its pose, they are object centered shape descriptions.

Correspondences are established between oriented points by comparing spin-images. If spin-images are represented as a set of 2-D points then comparisons will have to be made between points sets: a costly and ill-defined operation. Instead, as explained below, spin-images are represented as images that are compared through correlation.

To create the spin-image for the oriented point O on the surface of an object $\mathcal{M}$, the following procedure is invoked. For each point $\boldsymbol{x}$ on the surface of the object, the spin-map coordinates (α,β) with respect to O are computed. Next, the pixel P that the coordinates index in the image is determined by discretizing (α,β). Finally, the array is updated by incrementing the pixels surrounding *P* in the image. In order to blur the position of the point in the histogram to account for noise in the data and the discrete sampling of the surfaces in the scene, the contribution of the point is bilinearly interpolated to the four pixels surrounding (α,β). In general, the pixel size is set to two times the resolution of the surface mesh (measured as the average of the edge lengths in the mesh). Figure 2 shows some spin images for a CAD object. The darker the pixel, the more points have fallen into that particular bin.

The idea of spin-images evolved from concepts used in geometric hashing. Our initial idea was to use all of the points on the surface of an object in a basis geometric hashing algorithm. Unfortunately, constructing coordinate systems from all tuples of points would lead to a combinatoric explosion in the indexing [11] (given the large number of points on the surface). Furthermore, coordinate systems constructed from tuples of points are very sensitive to the position of points sensed on the surface. Instead, we decided to encode the position of points with respect to a 2-D basis defined with one oriented point, in order to reduce the combinatoric explosion and position sensitivity. After matching points by using a hash table, we determined that it would be just as effective, and much more efficient, to simply store an image that described the location of other points with respect to the oriented point, instead of performing lookup in a hash table. From this, the concept of a spin-image was born. Using spin-images to match points opens up the entire field of image based matching, giving us powerful comparison tools such as image correlation.

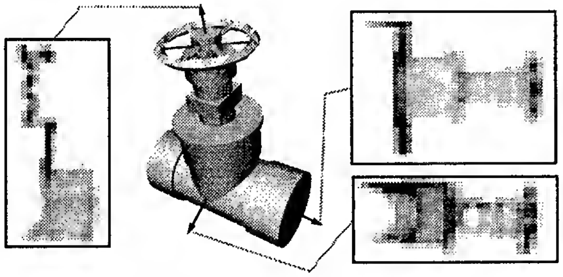


Figure 2: Some example spin images generated for three different oriented points on a CAD model of a valve.

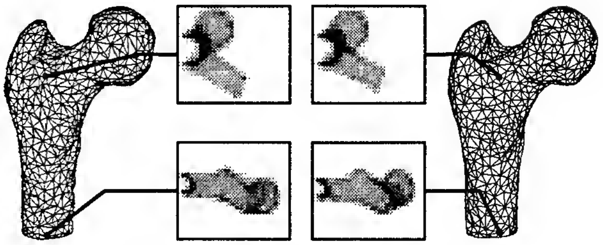


Figure 3: Spin-images generated from two different samplings of a model of a femur. Although the samplings are different, the spin-images generated from corresponding points are similar.

Because a spin-image is a global encoding of the surface, it would seem that any disturbance such as clutter and occlusion would prevent matching. In fact, this representation is resistant to clutter and occlusion, assuming that some precautions are taken. This will be described in detail in Section 4..

3. Comparing Spin-Images

Spin images generated from the scene and the model will be similar because they are based on the shape of objects imaged. However, they will not be exactly the same due to variations in surface sampling and noise from different views. For example, in Figure 3 the vertex positions and connectivity of two models of a femur are different, yet the spin-images from corresponding points are similar. A standard way of comparing linearly related images is the correlation coefficient. Because the correlation coefficient can be used to rank point correspondences, correct and incorrect correspondences can be differentiated.

The linear correlation coefficient provides a simple way to compare two spin-images that can be expected to be similar across the entire image. In practice, spin images generated from range images will have clutter (extra data) and occlusions (missing data). A first step in limiting the effect of clutter and occlusion, is to compare spin images only in the pixels where both of the images have data. In other words, the data used to compute the linear correlation coefficient is taken only from the region of overlap between two spin images. In this case, knowledge of the spin-image generation process is used to eliminate outliers in the correlation computation.

Since the linear correlation coefficient is a function of the number of bins used to compute it, the amount of overlap will have an effect on the correlation coefficients obtained. The more bins used to compute the correlation coefficient, the more confidence there is in its value. The variance of the correlation coefficient is included in the calculations of the relative similarity between two images so that the similarity measure between pairs of images with differing amounts of overlap can be compared. An appropriate similarity function C which we use instead of the correlation coefficient to compare spin-images P and Q where N is the number of overlapping bins is

$$C(P,Q) = (\operatorname{atanh}(R(P,Q)))^2 - \lambda\left(\frac{1}{N-3}\right) \qquad (2)$$

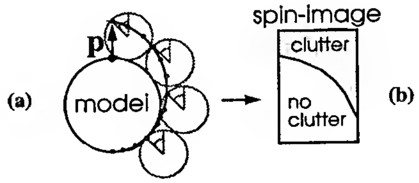
The similarity function will return a high value for two images that are highly correlated and have a large number of overlapping bins. The change of variables, a standard statistical technique ([4] Chapter 12) performed by the hyperbolic arctangent function, transforms the correlation coefficient into a distribution where the variance is independent of the mean. In (2), λ is a free variable used to weight the variance against the expected value of the correlation coefficient. In practice λ is set to three.

4. Limiting the Effect of Clutter and Occlusions

In real scenes, clutter and occlusion are omnipresent. Any object recognition system designed for the real world must somehow deal with clutter and occlusion. Some systems perform segmentation before recognition in order to separate clutter from interesting object data. In our case, the effects of clutter are manifested as a corruption of the pixel values of spin-images generated from the scene data. To some extent, the effect of clutter and occlusion can be limited by setting two thresholds that determine which points contribute to spin-image generation. The first threshold sets the maximum distance between the oriented point basis and a point in the mesh contributing to the spin-image. This parameter localizes the support of the spin-image to a sphere around its oriented point. In general this distance threshold is set to the size of the model (average distance of points on the model from its centroid). The second threshold sets the maximum angle between the oriented point basis surface normal and the surface normal of other points on the surface. This threshold prevents most points that will be self-occluded from contributing to the spin-image without specifying a viewing direction. This angular threshold is usually set to 90 degrees.

In order to analyze the effects of clutter, we have developed a simple model of the effect of clutter on spin-images under the assumption that objects are spherical. The clutter model combines the angular and distance thresholds explained above with the fact that objects of non-zero thickness cannot intersect to show that clutter is limited to connected regions in spin-images. Because of limited space, we cannot include a derivation of the clutter model, but our approach to clutter analysis is sketched in Figure 4. We are currently extending the clutter model to arbitrary objects by setting the radii of the spheres in the clutter model appropriately. Similar reasoning shows that the effect of occlusion is also limited to connected regions in spin-images.

Clutter and occlusion manifest as extra and missing points in the scene where the number of these points is bounded. Therefore, it is reasonable to assume that the total change of any pixel in a scene spin-image δ_i that is corrupted is bounded $|\delta_i| \le \delta$. Let the number of corrupted pixels in the scene spin-image be N_C and the total number of pixels be N. If the model and scene pixel values are normalized on [0,1], then the lower bound on the correla-



$$\text{(c)} \quad \rho_{LB} = \frac{\left(\sigma_m^2 - \frac{N_C}{N}\delta\right)}{\left(\sigma_m\sqrt{\sigma_m^2 + \frac{N_C}{N}(\delta + \delta^2)}\right)}$$

Figure 4: Theoretical clutter model. Because objects cannot intersect (a), the corruption due to clutter is limited to connected regions in spin-images (b). This results in a lower bound on the effect of clutter on the correlations coefficient of two spin-images (c).

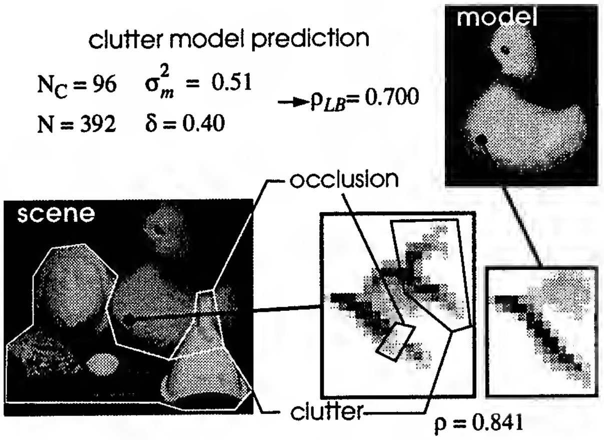

clutter model prediction
$N_C = 96$ $\sigma^2_m = 0.51$
$N = 392$ $\delta = 0.40$
$\rightarrow \rho_{LB} = 0.700$
model
occlusion
scene
clutter
$\rho = 0.841$

Figure 5: Experimental verification of clutter model. 96 of 392 pixels in the scene spin-image are corrupted by an amount δ less than 0.40. The correlation coefficient for the two images (0.841) is well above the lower bound (0.700) predicted by the clutter model. In the scene, the white lines indicate clutter and occlusion of the model and are not part of the original data.

tion coefficient when comparing model and scene spin-images is

$$\rho_{LB} = \left(\sigma^2_m - \frac{N_C}{N}\delta\right) \Big/ \left(\sigma_m \sqrt{\sigma^2_m + \frac{N_C}{N}(\delta + \delta^2)}\right) \qquad (3)$$

where σ^2_m is the variance of the pixels in the model spin-image. Hence, the worst case effect of clutter and occlusion grows sub-linearly with the area of corruption in the scene spin-image. Since clutter and occlusion cannot corrupt an entire spin-image and the effect of the corruption on the correlation coefficient is bounded, it can be concluded that matching of spin-images is only moderately affected by clutter and occlusion. Figure 5 validates our clutter model using spin-images from a a real scene with clutter and occlusion.

5. Generating Point Correspondences

The similarity measure (2) provides a way to rank correspondences so that only reasonable correspondences are established. Before recognition (off-line), spin-images are generated for all points on the model surface mesh and stored in a **spin-image stack.** At recognition time, a scene point is selected randomly from the scene surface mesh and its spin-image is generated. The scene spin-image is then correlated with all of the images in the model spin-image stack and the similarity measures ((2)) for each image pair are calculated and inserted in a histogram. As explained below, the images in the model spin-image stack with high similarity measure when compared to the scene spin-image produce model/scene point correspondences between their associated oriented points. This procedure to establish point correspondences is repeated for a random sampling of scene points that adequately cover the scene surface. Depending on the complexity and amount of clutter on the scene, this number can vary between one tenth and one half of the points in the scene. The end result is a list of model/scene point correspondences that are then filtered and grouped in order to compute transformation from model to scene.

Possible corresponding model points are chosen by finding the upper outliers in the histogram of similarity measures for each scene point. This method of choosing correspondences is reliable for two reasons. First, if no outliers exist, then the scene point has a spin-image that is very similar to all of the model spin-images, so definite correspondences with this scene point should not be established. Second, if multiple outliers exist, then multiple model points are similar to a single scene point, so should be considered in the matching process. We use a standard method for detection of outliers in a histogram ([4] Chapter 1); correspondences that have similarity measures that are greater than the upper fourth plus three times the fourth spread of the histogram are statistical outliers. Figure 6 shows a similarity measure histogram with detected outliers.

During matching, a single point can be matched to more than one point for two reasons. First, symmetry in the data and in spin image generation may cause two points to have similar spin-images. Second, spatially close points may have similar spin-images. Furthermore, if an object appears multiple

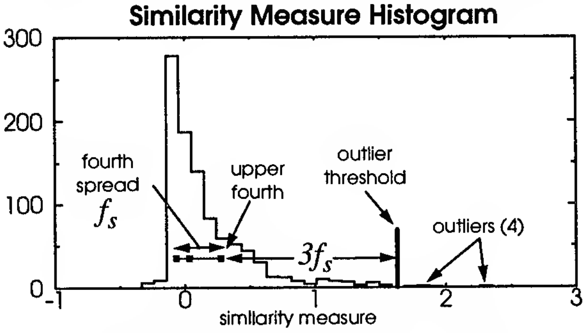


Figure 6: Similarity measure histogram.

times in the scene, then a single model point will match multiple scene points.

During matching, some points selected from scene clutter may be incorrectly matched to model points. However, given the numerous correspondences, it is possible to reason about which correspondences are actually on the model based on properties of the correspondences taken as a group. This *integral* approach is robust because it does not require reasoning about specific point matches to decide which correspondences are the best. This approach is in contrast to hypothesize and test and alignment paradigms of recognition where the minimal number of correspondences required to match model to scene are proposed and then verified through some other means.

First, similarity measure is used to remove unlikely correspondences. All correspondences with similarity measures that are less than some fraction of the maximum similarity measure of all of the correspondences are eliminated. In practice, this fraction is set to one half.

The second method for filtering out unlikely correspondences uses geometric consistency which is a measure of the likelihood that two correspondences can be grouped together to calculate a transformation of model to scene. If a correspondence is not geometrically consistent with other correspondences, then it cannot be grouped with other correspondences to calculate a transformation, and it should be eliminated.

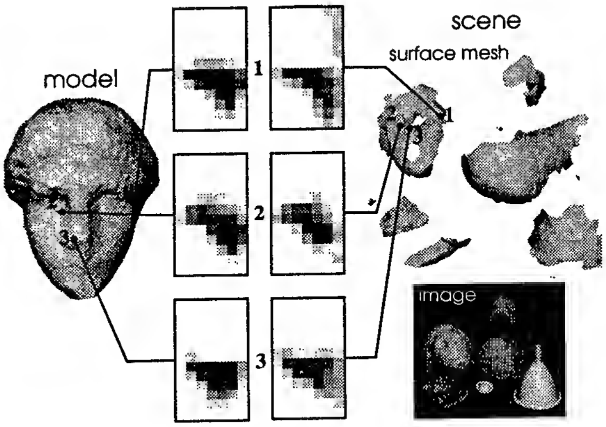


Figure 7: Three scene points and their best matching model points shown with associated best matching spin-images for a scene containing a model of the head of Venus.

The geometric consistency of correspondences $C_1 = [s_1,m_1]$ and $C_2 = [s_2,m_2]$ is measured by comparing the spin-map coordinates ((1)) of corresponding points.

$$d_{gc}(C_1, C_2) = 2\frac{\|S_{m_2}(m_1) - S_{s_2}(s_1)\|}{\|S_{m_2}(m_1) + S_{s_2}(s_1)\|} \qquad (4)$$

$$D_{gc} = max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1))$$

Normalized distance d_{gc} between spin-map coordinates is used because it is a compact way to measure the consistency in position and normals. Since d_{gc} is not symmetric, the maximum of the distances is used to define the geometric consistency distance D_{gc}.

To filter out correspondences based on geometric consistency, correspondences that are not geometrically consistent with at least one quarter of the correspondences in the list are eliminated. The end result after filtering on similarity measure and geometric consistency is a list of correspondences that are the most likely to be correct. In practice this number is between 20 and 50 correspondences. Figure 7 shows three of the best correspondences and matching spin-images between a model of the head of the goddess Venus and a scene containing it. The next step is to group these correspondences into sets that can be used to compute transformations.

6. Grouping Correspondences

Single correspondences cannot be used to compute a transformation from model to scene because an oriented point basis encodes only five of the six necessary degrees of freedom. At least two oriented point correspondences are needed to calculate a transformation if position and normals are used. To avoid combinatoric explosion, geometric consistency is used to cluster the correspondences into a few groups from which plausible transformations are computed. Since many correspondences are grouped together and used to compute a transformation, the resulting transformation is more robust than one computed from a few correspondences.

We cluster correspondences based on a measure of geometric consistency W_{gc} that is the geometric consistency distance between two correspondences (4) augmented by a weight that promotes clustering of correspondences that are far apart.

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - e^{\left\| S_{m_2}(m_1) + S_{s_2}(s_1) \right\| / 2}} \tag{5}$$

$$W_{gc}(C_1, C_2) = max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1))$$

W_{gc} will be small when two correspondences are geometrically consistent and far apart. The measure of geometric consistency between a correspondence C and a cluster of correspondences $\{C_i, \ldots, C_n\}$ is

$$W_{gc}(C, \{C_1, \ldots, C_n\}) = \max_i (W_{gc}(C, C_i)) \tag{6}$$

Given a list of correspondences $L = \{C_1, \ldots, C_n\}$, the clustering procedure for each correspondence is as follows: Select a seed correspondence C_i in L and initialize a cluster $G_i = \{C_i\}$. Find the correspondence C_j in L, for which $W_{gc}(C_j, G_i)$ is a minimum. Add C_j to G_i if $W_{gc}(C_j, G_i) < T_{gc}$ where the threshold T_{gc} is set to the size of the model. Repeat until no more correspondences can be added to G_i. The clustering procedure is performed for each correspondence in L, and the end result is n clusters, one for each correspondence in L. This clustering algorithm allows a correspondence to appear in multiple clusters which is necessary to handle model symmetry. For example, the CAD model in Figure 2 has a plane of symmetry resulting in two feasible transformations. Correspondences along the plane of symmetry contribute to two distinct transformations.

A plausible transformation T from model to scene is calculated from each cluster $\{[m_i, s_i]\}$ of correspondences by minimizing

$$E_T = \sum \left\| s_i - T(m_i) \right\|^2 . \tag{7}$$

Instead of using points and normals in the minimization of (7), we use only the position of the oriented points. This allows us to use a well defined algorithm for finding the best rigid transformation that aligns two point sets [6]. The transformations and associated correspondence clusters are then input into a verification procedure.

7. Verification

The purpose of verification is to find the best transformation of model to scene by eliminating matches that are inconsistent when all of the scene data is compared to all of the model data. Our verification algorithm is a formulation of the iterative closest point algorithm [1] [16] that can handle partially overlapping point sets and arbitrary transformations because is initialized with a transformation generated from correspondences determined by matching of spin-images.

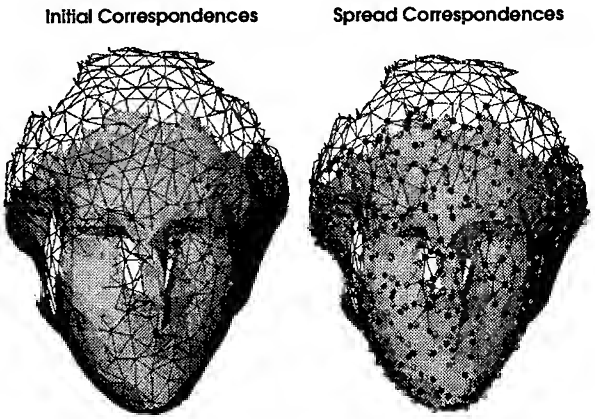


Figure 8: During verification, initial correspondences are spread over two views (one wireframe, the other shaded). Correspondences are prevented from being established outside the overlap of the views.

Verification starts with an initial set of point correspondences from which the transformation of model to scene is computed and then applied to the model points. For each correspondence, new correspondences are established between the nearest neighbors of the model point and nearest neighbors of the corresponding scene point if the distance between closest points is less than a threshold D_v. By finding scene points that are close to model points, this step grows the correspondences from those correspondences already established. The transformation based on the new correspondences is computed and then refined using traditional ICP. The growing process is repeated until no more correspondences can be established.

The threshold D_v in the verification stage (that sets the maximum distance by which two points can differ and still be brought into correspondence) is set automatically to two times the resolution of the meshes. This threshold allows for noise but prevents establishment of correspondences in regions where the data sets do not overlap.

Our algorithm grows patches of correspondences between the model and the scene from the initial correspondences and a cascade effect occurs. If the transformation is good, a large number of points will be brought into correspondence; if the transformation is bad, the number of correspondences will remain close to the original number. Therefore, a good measure of the validity of the match is the number of correspondences after verification. Since a large number of correspondences are used to compute the final transformation, the alignment will be more accurate than that computed through matching of spin-images.

Figure 8 illustrates how initial correspondences, established by matching spin-images, are spread over the surfaces of two range views of a plastic model of the head of the goddess Venus. The correspondences are established only in the regions where the two surface meshes overlap, thus preventing a poor registration caused by correspondences being established between non-overlapping regions.

8. Multiple Models

A strong property of our recognition algorithm is that it permits simultaneous recognition of multiple models. Recognition with multiple models is similar to recognition with one model except that each scene point is compared to the spin-images of all of the models. The rest of the algorithm is the same except that correspondences with model points from different models are prevented from being clustered.

Figure 9 demonstrates the simultaneous recognition of two models of free-form shape. The femur and pelvis model were acquired using a CT scan of the bones, followed by surface mesh generation from contours. The scene was acquired using a K^2T structured light range camera. Both the models and scene data were processed by removing long edges associated with step discontinuities, applying a "smoothing without shrinking" filter [14], and then applying a mesh simplification algorithm that preserves the shape of objects in the scene while evenly distributing the points over the its surface [8]. Our algorithm was able to recognize the objects even in the presence of extreme occlusion; only 20% of the surface of the pelvis is visible. This result also demonstrates the recognition of objects sensed with different sensing modalities.

9. Results

Our main application domain is interior modeling. In interior modeling, objects are recognized in range images of complex industrial interiors. By recognizing objects, a semantic meaning is associated with the objects in the scene, setting the stage for high-level robotic interaction. For example, by recognizing a valve in the scene, a robot can be given a high-level commands such as "turn off the valve" [9].

Figure 10 shows the result of recognizing four different industrial objects in cluttered industrial scenes. The surface mesh models were generated by CAD drawings using finite element software to tessellate the surface of the objects. The scene images were acquired with a Perceptron 5000 scanning laser rangefinder. Before recognition, the

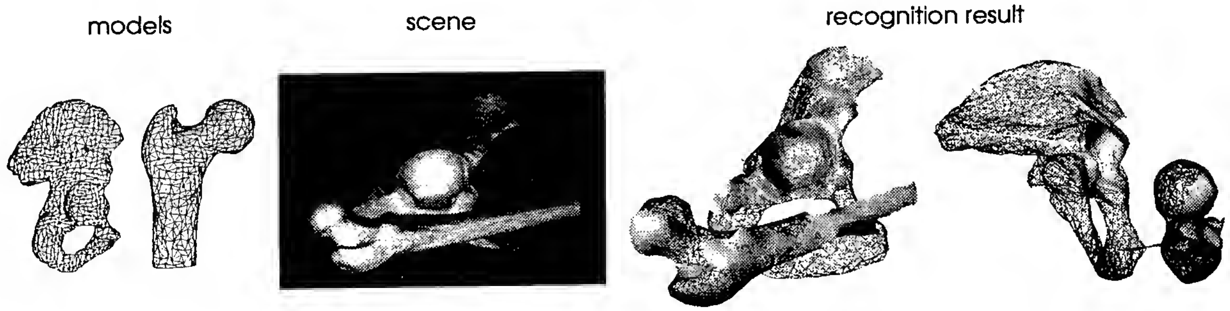


Figure 9: The simultaneous recognition of the models of a femur and a pelvis bone in a range image. The femur and pelvis are recognized even in the presence of extreme occlusion (only 20% of the pelvis surface is visible) and clutter. From left to right are shown the pelvis and femur models acquired with CT, the scene image from a structured light range camera, and then two views of the recognition results where the scene data is shaded and the models are in wireframe. This result demonstrates simultaneous recognition of free-form objects acquired using different sensing modalities.

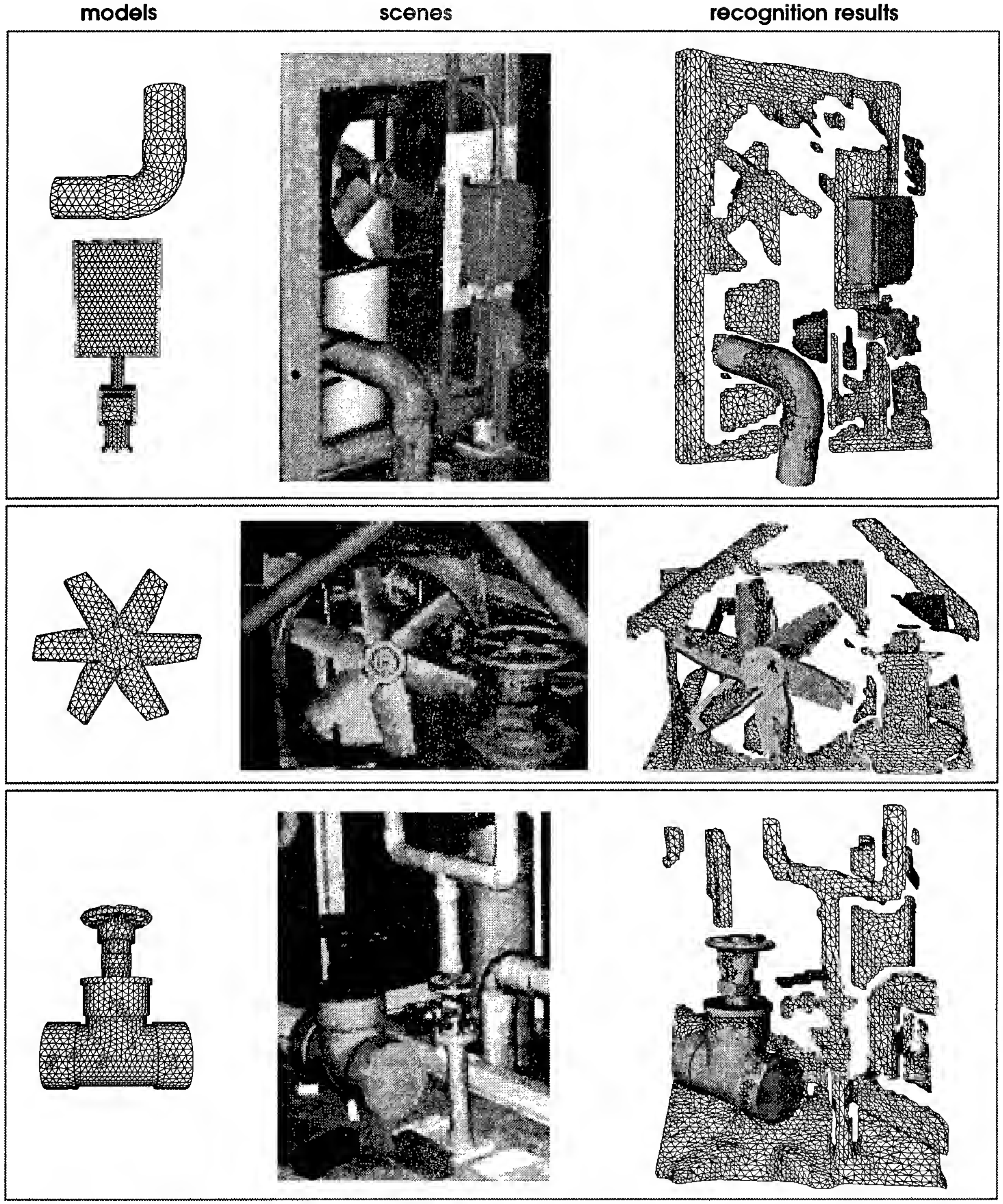


Figure 10: The recognition of industrial objects in complex scenes. On the left are shown wireframe models which were created from CAD drawings using finite element software for surface tessellation. In the middle are shown the intensity images acquired when a scanning laser rangefinder imaged the scene. On the right are shown the recognized models (shaded) superimposed on the scene data (wireframe). These results demonstrate the recognition of complicated symmetric objects in 3-D scene data containing extreme clutter and occlusions. All of the scene data points are used in the recognition and no object/ background segmentation is performed.

scene data is processed to remove long edges and small surface patches, smoothed and simplified. In all examples, the scene data is complex with a great deal of clutter. Furthermore, all the models exhibit symmetry which makes the recognition more difficult, because a single scene point can match multiple model points.

In addition to these results, we have generated results from multi-view merging and alignment of terrain maps [7].

10. Future Work

We have presented a recognition algorithm that is based matching spin-images generated using oriented points on the surface of an object. Although spin-images are global representations, they are robust to clutter and occlusions. We have demonstrated our algorithm's ability to handle clutter and occlusions through recognition of objects in complex cluttered scenes.

In the future, we will extend the algorithm to recognize multiple objects simultaneously from a library of models. This will require efficient methods for determining which models in the library are present in the scene. We have determined that all of the spin-images for a model lie on a 2-D manifold in the high-dimensional spin-image space. It is likely that, through the use of eigen-space compression, this manifold can be projected onto a much lower dimensional manifold. Rapid determination of which models appear in the scene could then be determined by projection of scene spin-images onto a manifold generated for each model in the library. This eigen-spin-image approach is similar to parametric appearance based matching.

References

[1] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239-256, 1992.

[2] R. Bergevin, D. Laurendeau and D. Poussart, "Registering range views of multipart objects," *Computer and Vision Image Understanding*, 61(1):1-16, 1995.

[3] C. Chua and R. Jarvis, "3-D free-form surface registration and object recognition," *Int'l J. Computer Vision*, 17(1):77-99, 1996.

[4] J. Devore, *Probability and Statistics for Engineering and Sciences*, Brooks/Cole, Belmont, CA, 1987.

[5] A. Guéziec and N. Ayache, "Smoothing and matching of 3-D space curves," *Int'l J. Computer Vision*, 12(1):79-104, 1994.

[6] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Optical Soc. Amer.*, 4(4):629-642, 1987.

[7] A. Johnson and M. Hebert, "Recognizing objects by matching oriented points," *CMU Robotics Institute TR, CMU-RI-TR-96-4*, May 1996.

[8] A. Johnson and M. Hebert, "Control of mesh resolution for 3-D object recognition," *CMU Robotics Institute TR, CMU-RI-TR-96-20*, December 1996.

[9] A. Johnson, P. Leger, R. Hoffman, M. Hebert, J. Osborn, "3-D object modeling and recognition for telerobotic manipulation," *Proc. Intelligent Robots and Systems 1995 (IROS '95)*, pp. 103-110, August 1995.

[10] K. Ikeuchi, T. Shakunaga, M. Wheeler and T. Yamazaki, "Invariant Histograms and deformable template matching for SAR target recognition," *Proc. Computer Vision and Pattern Recognition (CVPR 1996)*, pp. 100-105, 1996.

[11] Y. Lamdan and H. Wolfson, "Geometric Hashing: a general and efficient model-based recognition scheme," *Proc. Second Int'l Conf. Computer Vision (ICCV '88)*, pp. 238-249, 1988.

[12] F. Pipitone and W. Adams, "Tripod operators for recognizing objects in range images; rapid rejection of library objects," *1992 IEEE Robotics and Automation (R&A 1992)*, pp. 1596-1601, 1992.

[13] F. Stein and G. Medioni, "Structural Indexing: efficient 3-D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2): 125-145, 1992.

[14] G. Taubin, "A Signal processing approach to fair surface design," *Proc. Computer Graphics 1995 (SIGGRAPH '95)*, pp. 351-358, 1995.

[15] J. Thirion, "New feature points based on geometric invariants for 3D image registration," *Int'l J. Computer Vision*, 18(2):121-137, 1996.

[16] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int'l J. Computer Vision*, 13(2):119-152, 1994.

Acknowledgments

This research was performed as part of the Artisan Project in the Field Robotics Center and was funded by the Department of Energy under contract DE-AC21-92MC2910. We would like to thank Jim Osborn for encouraging this work and David Simon for providing the CT data sets.

MEDIALNESS AND SKELETONIZATION FOR OBJECT REGISTRATION AND SHAPE SIMILARITY

Andrew Bzostek
Lawrence B. Wolff
Computer Vision Laboratory
Department of Computer Science
The Johns Hopkins University
Baltimore, Maryland 21218

Abstract

A new computational approach to the concept of medialness and skeletonization of a grey level image set is presented, motivated by the original notion of the Blum "Grassfire" Transform definition for the skeleton of a binary set. A dynamic programming algorithm generates a unit vector field in the interior of an image set surrounded by a boundary curve whose integral curves are the trajectories of the contracting boundary curve points. The positive divergence of this unit vector field is used as a measure of "medialness", while negative divergence of the vector field with same direction but whose length is the local grey level gradient measures "boundariness". Presented are preliminary results for generating medialness, skeletons and using these for registration and measuring shape similarity.

1 Introduction

The Medial Axis Transform has enjoyed a long and rich history in the field of computer vision [2, 3, 4, 6, 5] particularly related to object representation in images. A number of definitions have been given for essentially the same transformation of an image set into its skeleton. One of the original definitions for the medial axis transform [1, 7] reduces an object in a binary image into a set of interior points each of which are at the center of a metric contour (e.g., a sphere for Euclidean metric) wholly contained in the object and intersecting the boundary of the object in two or more locations. Such a construction has been extended to grey level images and made robust to noise by using a multi-scale approach called "Cores" [6, 5]. Effectively the notion of medialness at a point in a grey level image at a given scale is the summation of edge strength along the metric contour centered at this point and whose radius is at the given scale: the "Core" is defined in terms of ridges of medialness in scale-space.

We propose a complementary approach to the notion of medialness which can provide augmented utility particularly for object registration and determination of shape similarity. This paper presents preliminary ideas and results in this direction. The intuitive basis for medialness that we present goes back to the "grassfire technique", directly quoting Blum and Nagel [4]

Imagine an object whose border is set on fire. The subsequent internal quench points of the fire represent the symmetric [medial] axis, the time of quench for unit velocity propagation being the radius function.

We extend this concept mathematically by assigning to each point in the object the unit vector direction that the fire is propagating as it passes over the point. Thus as the fire burns it defines a unit vector field across the object. For a uniform object the integral curves for this vector field are simply the straightline trajectories of the fire's progression that eventually run into each other at the quench points. At the quench points unit vectors are discontinuously changing in direction: hence for us the medial axis arises from singularities in this unit vector field. To go further, we study the measure of *medialness* from the computation of divergence of this vectorfield, and ridge lines of divergence form our notion of the skeleton of an object.

The extension to grey level images follows naturally by interpreting grey value as density where higher grey values impede more the progress of the fire from the boundary towards the interior of the object. This leads to unit vectors that trace out paths of minimal integrated density from the boundary to the interior. In general the boundary from which the vectorfield starts need not be of an object, it can be the border of an arbitrarily shaped *aperture* placed over an image region. The integrated density (i.e., integrated grey value) defines a height function inside the aperture, and the unit vector field of interest is parallel to the gradient vector for this height function. In the special case of an aperture whose border is aligned with the boundary of a uniform grey object this height function is simply the value of the Euclidean Distance Transform. By computing the divergence of the gradient vectorfield of the height function (as opposed to the unit vector field) boundary information gets computed from changing grey level information. In combination with medialness information from the divergence of the unit vector field, this provides important information for distinguishing exterior/interior of objects in a grey level image.

We show various examples of computing divergence for vectorfields for grey level objects demonstrating the computation of medialness and in some cases boundaries. We then show how this vectorfield can be used for similarity matching, and how the singularities of the vectorfield (i.e., ridges of divergence forming the skeleton) can be used to perform an initial coarse object registration, with the rest of the vectorfield used to refine this registration.

2 Some Basic Examples

Given a density image and an aperture, a height function can be defined at each point interior to the aperture as the integrated density (ID) along the path of from the

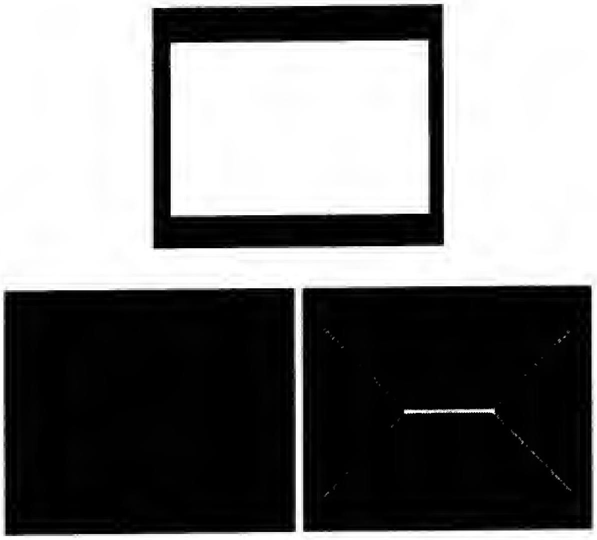


Figure 1: (a) Top image shows uniform grey level Rectangle, (b) Left image shows interior vectorfield, (c) Right image shows divergence of interior vectorfield (middle grey = 0 divergence, light grey = positive divergence, dark grey = negative divergence.

point to the boundary of the aperture with minimum ID. We generate both this height function and its gradient at each location. In the discrete case, this is calculated for each pixel using a dynamic programming algorithm. At each point, the vector to an 8 neighbor is chosen which minimizes the ID of this vector plus the height at the neighbor point. Using a priority queue sorted by height, these values are calculated in O(n logn) time.

Figure 1a shows a uniform grey level rectangle against a dark background. Figure 1b shows the unit vector field generated from the inward propagation of a fire set at the boundary of the rectangle, using the dynamic programming algorithm described above. Figure 1c shows a grey level representation of the local divergence of the vectorfield in Figure 1b by performing convolution with the 3x3 kernel shown in Figure 2. The computation of convolution is achieved by summing each of the respective dot products of a vector in the kernel with the vector in the image it overlays. The brighter grey values in Figure 1(c) show the ridges of divergence values depicting the skeleton. Unlike a binary skeleton, the divergence value at each point on the skeleton gives the "medialness" at that skeleton point.

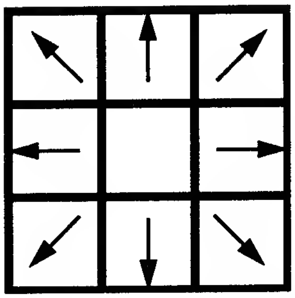


Figure 2: Convolution kernel for divergence computation.

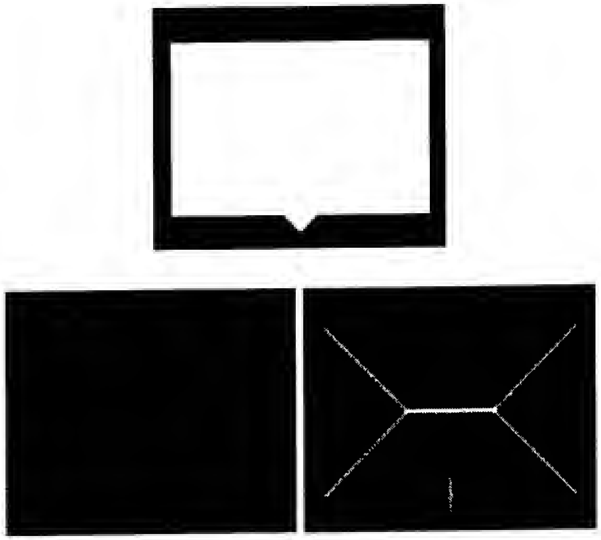


Figure 3: (a) Top image shows uniform grey level Rectangle with triangle appended to bottom edge, (b) Left image shows interior vectorfield, (c) Right image shows divergence of interior vectorfield (middle grey = 0 divergence, light grey = positive divergence, dark grey = negative divergence).

Figure 3a shows a grey level rectangle with a small triangle appended to one of its sides. Figure 3c shows the resulting divergence image and the additional appendage to the skeletal ridge as a result of the additional triangle. While the skeleton has an additional segment, the divergence (i.e., "medialness") along this ridge is much smaller further away from the triangle. If instead a triangle of smaller size were appended (e.g., on the order of size of noise to the shape of the boundary of the rectangle), the divergence becomes neglible except in the close vicinity of the noise. In effect, for noise along the boundary shape, the main skeletal ridge for the rectangular shape remains the same with boundary noise producing disconnected skeletal ridges only in the spatial vicinity of the noise.

Figure 4a shows a rectangle composed of two rectangles each of two significantly different uniform grey values. Figure 4b shows the interior vectorfield generated for an aperture surrounding both rectangles together. Figure 4c shows that the resulting computation of divergence creates two disconnected ridges of positive divergence. Separating these two ridges of positive divergence is a dark "boundary" of negative divergence. Although preliminary, this phenomenon shows how objects of different grey value composition can be potentially segmented by looking at ridges of positive and negative divergence.

The left of Figure 5 shows a grey level image for a set of objects, and the right of Figure 5 shows divergence computations for the interiors of each object with aperture respectively being at the boundary for each object. The next section shows an example of registration between two objects in these images (i.e., the rectangular shapes with three holes) using the skeletal ridge and interior vector field.

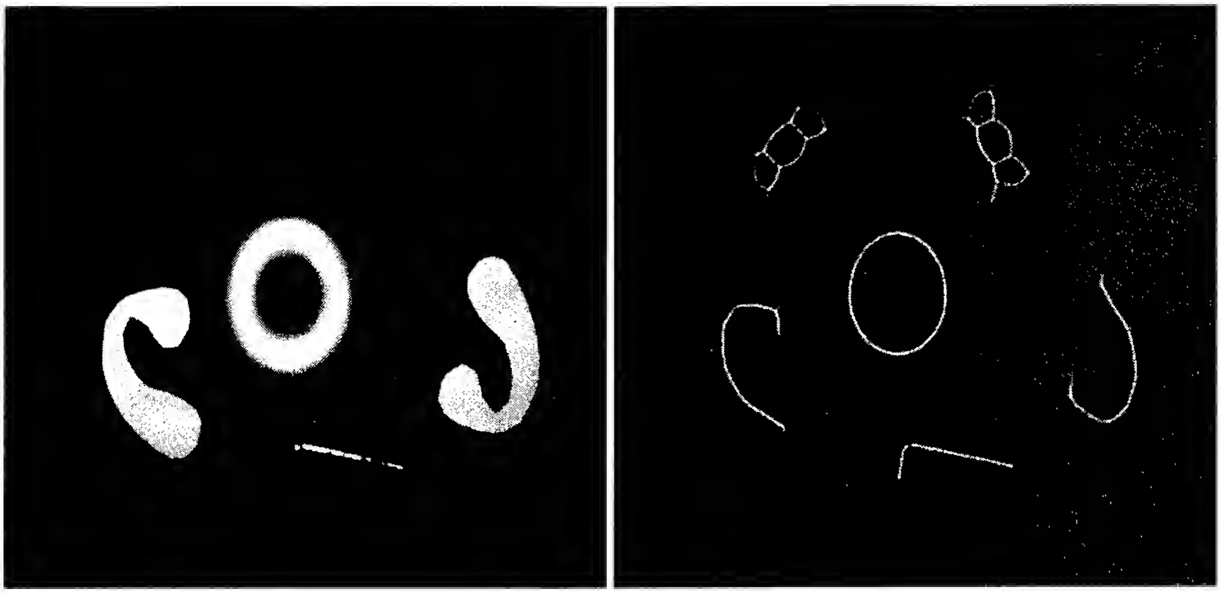


Figure 5: (Left), a grey level image of an assortment of shapes, (Right), divergence computed for the interior vectorfield of each grey level object.

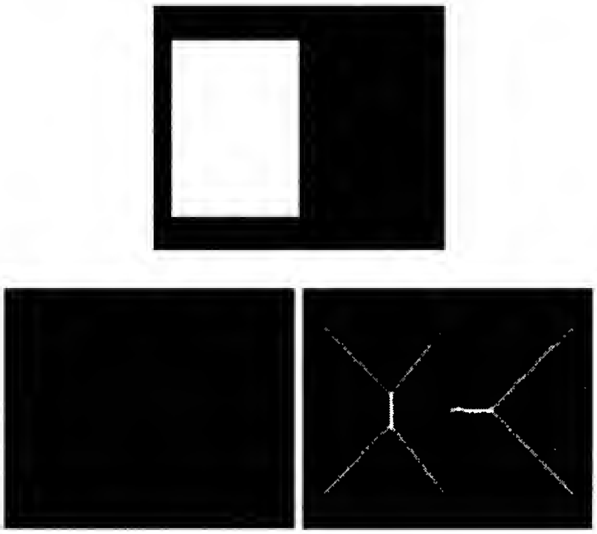


Figure 4: (a) Top image shows uniform grey level Rectangle with triangle appended to bottom edge, (b) Left image shows interior vectorfield, (c) Right image shows divergence of interior vectorfield (middle grey = 0 divergence, light grey = positive divergence, dark grey = negative divergence).

3 Registration and Similarity Between Objects

Figure 6 and Figure 7 show closeups of the objects and respective interior vectorfields and divergence computations, to be registered.

Registration of two objects is achieved in four steps. First, the skeletons of images are computed, taken as the largest connected set of ridges in the images. Each of these is then collapsed into a graph whose nodes have degree != 2. Associated with each connected pair of nodes is a curve of the skeleton, represented as a set of points. Any terminal curve (ending with node degree = 1) whose average divergence is not above a threshold set as a percentage of the average divergence of all of the skeleton curves is removed and the skeleton further collapsed. A set of topology-preserving matchings between the set of nodes for each is generated in a brute-force manner. The match with minimum squared difference between the lengths of matched curves is then chosen. The rigid transformation, consisting of a combination of 2-D translation and rotation, which minimizes the squared distances between matched nodes is calculated. This transformation is applied to all further positions and vectors from the first image.

Next, after this initial coarse registration, the matched curves are then parameterized by integrated divergence. Each pixel along the curve in the image to be transformed is matched with a location on the associated curve in the other. This location may be non-integral, but because we will be calculating the transformation for each pixel only in the second image, this is OK. The pixels in the second image can be partitioned into two sets, based on whether or not they have vector incident on them, i.e. whether or not any other pixel's vector points to them. In addition to the pixels on the skeleton, all adjacent pixels which do not have vectors incident on them are also mapped, adjusted by distance from the skeleton point scaled by the ratio of the divergences, normalized by total divergence along the curve. Then, for each matched point on or near the skeletons, the path in the vector field for which it is a source is followed, stepping by pixel in the second image. At each step, the previous matched location in the original image is modified by its current, interpolated vector, scaled by the ratio of the vector lengths in the two images normalized by the total path length. Finally, the rest of the non-target set of pixels is taken in descending order of divergence. For each, its path is followed, finding the corresponding locations until, either an already mapped pixel is found, or the boundary is reached. After all of the points in the non-incident set have been investigated, all pixels in the second image have associated with them

Figure 6: Left rectangle shape with 3-holes from Figure 5.

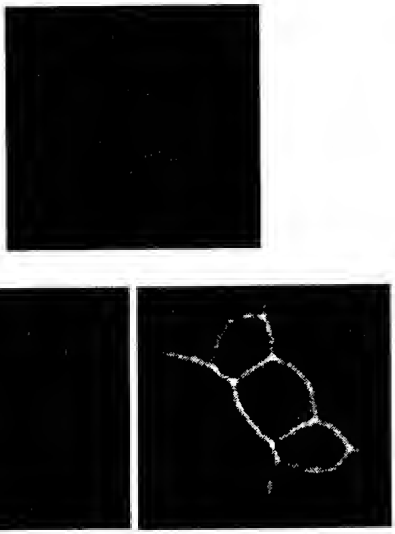


Figure 7: Right rectangle shape with 3-holes from Figure 5.

a location in the first, as well as a vector. The locations represent the non-rigid residual transformation after the rigid transformation is applied.

Two proposed measures for accuracy of registration, as well as shape similarity, using the interior vectorfields for two objects, are as follows:

$$\frac{1}{||obj1 \cap obj2||} \sum_{obj1 \cap obj2} (\hat{V}_{obj1} \cdot \hat{V}_{obj2})^2$$

$$\frac{1}{||obj1 \cap obj2||} \sum_{obj1 \cap obj2} ||\vec{V}_{obj1} \times \vec{V}_{obj2}||^2 \cdot ||\vec{V}_{obj1} - \vec{V}_{obj2}||^2 .$$

4 Conclusion

This paper has presented preliminary research using a complementary approach to the computation of medialness using divergence of an interior vectorfield to an object. Some basic examples of these computations were shown and outlined were methods for registration. Mesures for testing the accuracy of registration, as well as shape similarity, were proposed.

References

[1] D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1968.

[2] H. Blum. A transformation for extracting new descriptors of shape. In *in Models for the Perception of Speech and Visual Form*, pages 362–380, Cambridge, Massachusetts, MIT Press 1967.

[3] H. Blum. Biological shape and visual science (part i). *J. Theoretical Biology*, 38:205–287, 1973.

[4] H. Blum and R. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[5] C. Burbeck and S. Pizer. Object representation by cores: Identifying and representing primitive spatial regions. *Vision Res.*, 35(13):1917–1930, 1995.

[6] S.M. Pizer et al. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):505–511, 1987.

[7] A. Rosenfeld and A. Kak. *Digital Picture Processing*. Academic Press, 1982.

Hierarchical Silhouettes Classification using Curve Matching

Yoram Gdalyahu and Daphna Weinshall,
Institute of Computer Science, The Hebrew University,
91904 Jerusalem, Israel

Abstract

We developed an algorithm that achieves good and robust matching between two contours of possibly rather different shapes. We describe extensive experiments with real images of $3D$ objects, and demonstrate excellent matching results for curves under partial occlusion, for curves describing the same object observed from different viewpoints, and for curves describing different but related objects. Furthermore, we used the method to compare a range of curves taken from a large database of real images of various toy models. We then used the results to classify the data with an automatic hierarchical clustering algorithm, getting excellent results which faithfully captured the real structure in the data. This serves as indirect evidence to the quality of our matching algorithm.

1 Introduction

We discuss the problem of contour matching in the context of learning from visual examples. In a nutshell, we wish to reveal the structure which underlies a large collection of images and shapes; for this purpose we shall use hierarchical clustering, where shapes are hierarchically categorized according to their mutual similarities. Thus we need to accomplish reasonable matching between weakly similar curves. This is unlike typical model based recognition applications, where one need only determine whether the model curve and the image curve are the same, up to an image transformation (e.g., translation, rotation and scaling) and some permitted level of noise [1; 4; 5; 6]. The desired qualitative matching is also unlike other applications of curve matching, like stereo or tracking [3].

We do not have a precise definition for what amounts to a reasonable matching between weakly similar curves. As an illustrative example, consider two curves describing the shape of two different mammals: possibly we would like to see their limbs and head correspondingly matched. Thus it appears that we would like to match curve sections that are locally similar. We will therefore define a suitable heuristic measure for local similarity, and we will propose a matching algorithm that maximizes this measure.

There is a fundamental difference between local and global methods of curve matching. Our method is purely local, since we would like to apply it also in cases of occlusions, where partial matching is desired. Moreover, while many available local methods assume that the whole visible image curve can be matched with some portion of a pre-stored model curve, we avoid this assumption, which is justified for model based recognition applications, but not in the present study. Instead, our problem is to find subsections on two given curves, which are "sufficiently" similar.

We are motivated by an application that lacks a-priori knowledge, as in [7]. However, a great difference between the present work and [7] is that the later method is not local: it involves using the total length of the contour.

In the present work we use a segment representation, which is a polygonal approximation of the (closed) curve. Segments are matched in a fine-to-coarse multi-scaled approach. More specifically, we try to match the given curves at their maximal available resolution, but we allow for segment merging which is expected to remove the noise and fine resolution details. We transform the similarity optimization problem to a set of dynamic programming problems, which compete for the best solution.

In the next section we describe our matching algorithm. In section 3 we show matching results, and we also show an application to image classification. The perfect image clustering that is obtained stands as an evidence for the quality of our matching results.

2 The matching algorithm

Let us denote by $A = \{a_i\}$ the sequence of segments which represent a linear approximation of a curve. Each segment is defined by its length $\ell(a_i)$ and slope $\theta(a_i)$. We also define a special symbol $\ominus$ that can be inserted between two segments. The symbol $\ominus$ represents a discontinuity in the contour, or a "gap segment"; it does not have length or slope characteristics.

Image scaling is expressed in this representation by $\ell(a_i) \to C \cdot \ell(a_i) \;\; \forall i$ and image rotation is expressed by $\theta(a_i) \to C + \theta(a_i) \;\; \forall i$.

In addition to these two operations on A, we define the following operations as well:

- Interruption: the operation of inserting a gap element $\ominus$ into A.
- Merge: the operation of replacing m consecutive elements, which do not contain $\ominus$, by a single new element, that represents their vectorial sum.
- Shift: the operation of (cyclic) re-ordering, i.e., $a_i \leftarrow a_{i+1}\ \forall i$. For a closed curve with n segments $a_n \leftarrow a_1$, and for an open curve i can be zero or negative.

For example, the set $\{a_1, a_2, a_3, a_4, a_5, a_6\}$, which represents a closed curve, may be transformed to $\{a_3, a_{4+5}, a_6, a_1, \ominus, \ominus, a_2\}$ by two shift operations, one merge operation and two interruptions.

Next we define a measure for local similarity between contours. Let A and B represent two given contours. Then:

$$S(A,B) = \sum_{i=n_1}^{n_2} s(a_i, b_i) \qquad (1)$$

where $n_1 \leq n_2$ are valid indices in A and B, and $s(a_i, b_i)$ is a pairwise segment similarity function to be defined.

Given two contours which are represented by A and B, the goal of the matching algorithm is to transform them into A' and B' for which the local similarity $S(A', B')$ is maximal.

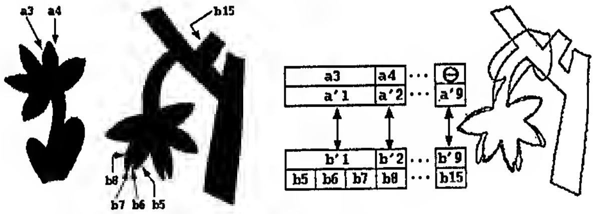


Figure 1: Matching of two sketches, drawn by hand. On the right: the alignment transformation which minimizes two thirds of the distances between the matched features (see text).

The motivation behind our formulation may be clarified with a simple example. Figure 1 shows two hand-drawn contours that are matched using our method. Segments $b_5 - b_7$ are merged into one segment (denoted b'_1), which is in turn matched to segment a_3 (denoted a'_1). Hence, in this case, the merging mechanism is used to compensate for noise or scale (resolution) differences. On the other hand, the gap mechanism ($\ominus$ insertion) is used to compensate for a long occluded sequence of features. For example, segment b_{15} lies within such a sequence, and it is therefore matched with a gap $\ominus$.

We will now discuss the similarity measure in more detail. The segment similarity function $s(a, b)$ from Eq. (1) is defined as

$$\begin{aligned} s(a,b) &= w_1\ s_\ell\left(\ell(a), \ell(b)\right) + s_\theta\left(\theta(a), \theta(b)\right) && (2) \\ s(a,\ominus) &= s(\ominus, b)\ =\ w_2 && (3) \end{aligned}$$

where $a, b \neq \ominus$, and $s(\ominus, \ominus)$ is not defined. In (2) we compute separately the scale similarity and orientation similarity of two segments a and b; we then let them contribute additively to the total segment similarity, with a relative weight of w_1. This choice is somewhat arbitrary, although it appeals to our intuition. Nevertheless, whichever way $s(a, b)$ is chosen, we require that it would be symmetric, i.e., $s(a, b) = s(b, a)$. We also require that w_2 in (3) be a constant.

Our specific choices for s_ℓ and s_θ are the following:

$$\begin{aligned} s_\theta(a,b) &= \cos\left(\theta(a) - \theta(b)\right) && (4) \\ s_\ell(a,b) &= \frac{2\ \ell(a)\ \ell(b)}{\ell^2(a) + \ell^2(b)} \ = \ \cos 2\delta && (5) \end{aligned}$$

where δ is the angle in R^2 between the vector $[\ell(a), \ell(b)]$ and the vector [1,1].

The value of $s(a, \ominus)$ is not intended to measure the similarity between a segment a and a gap segment $\ominus$, which is meaningless. Instead, w_2 is a matching threshold. When $s(a, b) < w_2$ it might be preferable to insert a gap element, and gain at least the value of $s(a, \ominus)$ or $s(\ominus, b)$. Increasing the value of w_2 implies that a higher level of similarity between matched segments is required, and the length of non interrupted matched sequences is consequently reduced.

So far, the total contribution of interruptions is determined only by the number of gap elements that have been inserted. However, the case of a few consecutive gap elements should lead to greater similarity as compared to the case when the same number of gap elements is uniformly spread along the contour. This is because the first case could arise from occlusion, whereas the second case arises typically from curve dissimilarity. To take this factor into account, we modify our previous definition of the contour similarity $S(A, B)$; we now add to the pairwise segment sum a penalty term, which reduces the similarity if the gap elements are not connected. Namely:

$$S(A,B) = \sum_{i=n_1}^{n_2} s(a_i, b_i) - N_G \cdot w_3 \qquad (6)$$

where N_G is the number of connected chains of gap elements, and w_3 is a weight factor.

In the maximization of (6) we separate between global operations, which act on the whole curve (rotation, scaling and segment enumeration shift) to local operations (interruption and merge). The global parameters define a three dimensional parameter space, over which the similarity criterion (6) should be maximized. We take a quite conventional approach, and seek an approximate solution among the $N \cdot M$ initial alignments[1] that map some segment of one curve to a segment of the other curve. Here N and M are the number of segments of A and B.

The choice of a particular initial alignment defines the three global parameters. For each of the $N \cdot M$ initial

[1] There is another factor of 2 if reflections are allowed.

alignments we start the evaluation of (6), where the optimal local operations are found by dynamic programming. More specifically, for two given ordered sets A and B, which are globally aligned and which consist of N and M elements respectively, we assign an array $R_{N\times M}$. The entry $R[i,j]$ holds the maximal similarity that can be achieved when the first i elements of A are matched with the first j elements of B. The entries of R are updated in a proper order according to the following rule:

$$R[i,j] = \max\{r_1, r_2, r_3\} \qquad (7)$$

where

$$\begin{aligned} r_1 &= \max_{\alpha,\beta\in\Omega}\{R[\alpha,\beta] + s(\overline{\alpha i}, \overline{\beta j})\} \\ r_2 &= \max_{0<\alpha<i}\{R[\alpha,j] + w_2\cdot(i-\alpha) - w_3\} \\ r_3 &= \max_{0<\beta<j}\{R[i,\beta] + w_2\cdot(j-\beta) - w_3\} \end{aligned}$$

$\overline{xy}$ denotes the vectorial sum of the segments $(x + 1), \ldots, y$, and the domain Ω is defined below.

The term r_1 denotes the similarity score that can be achieved by extending a previously obtained match. The extension may consist of a single segment addition to each curve, or the addition of a few segments which are merged together. The total number of added segments is bounded by a constant K, and therefore:

$$\Omega = \{\alpha, \beta \mid 0 < \alpha < i,\ 0 < \beta < j,\ (i-\alpha)+(j-\beta) \leq K\}.$$

Thus the domain Ω contains no more than $K(K-1)/2$ elements.

The terms r_2 and r_3 in equation (7) provide an alternative to continuous segment matching. The alternative is to use the interruption mechanism, and match ξ segments of one curve, with $\ominus$'s that are inserted into the other curve. This operation can contribute to $R[i,j]$ only the pre-defined quantity $w_2\cdot\xi - w_3$, and hence it will be chosen only in cases where the continuous matching is poor and can contribute even less.

When R is fully updated, the path in R, which arrives to the entry with maximal score, represents the optimal segment correspondence between the two curves.

In order to avoid the computation of all the dynamical arrays, we use two strategies. The first strategy makes the update process competitive: We define a potential function $f(R_l)$, which decreases monotonically during the process, and which gives an upper bound on the values that a given R_l may achieve when the process ends. In the field of AI, $f(R)$ is called "optimistic heuristic function". The process is transformed to a competitive one by choosing at each step a different array to be updated, namely, the array R_l for which $f(R_l)$ is maximal. At the completion of the update step, the value of $f(R_l)$ is decreased. Since $f(R)$ is based on optimistic estimation, the optimal solution cannot be missed.

The second strategy, which we use to reduce unnecessary computations, is statistical filtering. This strategy is complementary to the competitive strategy, since the competitive strategy is not effective, and even wasteful, in the initial stages of the update process. Nevertheless, after a few updating steps, we can already get a rough estimate of the global parameters, and continue the competition between the arrays that are not far from our estimate.

3 Results

3.1 Pairwise contour matching

Figure 2 demonstrates the local nature of the matching method. The difference in length between the outlines of the two silhouettes, which in this example results from occlusion, does not impede the correct matching of the common parts. This is due to the fact that our method does not require global image normalization. Note also that although the cow in the two images was photographed from slightly different points of view (note the distance between the front legs and the number of ears), the matching is essentially perfect.

Wrong matches cannot be avoided, but they are usually easy to identify since they typically correspond to large distances between the matched features. The reason behind this convenient characteristic of errors is that the matching algorithm depends only on the local shape of the curve. In figure 2 the mismatches were eliminated by a distance threshold (see the dashed lines).

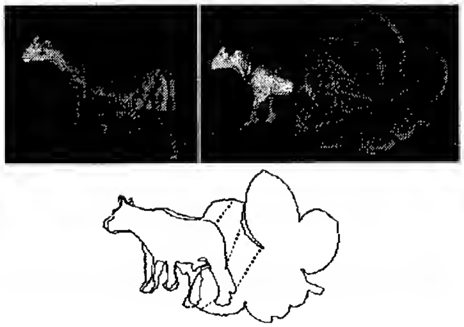


Figure 2: The alignment transformation which minimizes 2/3 of the distances between matched features. Four of the ignored matches are denoted by dashed lines. There were 64 and 40 features extracted from the two contours, of which 36 pairs were matched.

The next example (figure 3) illustrates the qualitative nature, or the flexibility, of our matching method. Recall that our intended application is classification, where it is necessary to compute the similarity between images of different objects.

The last example shows an application of the algorithm to match images taken from very different points

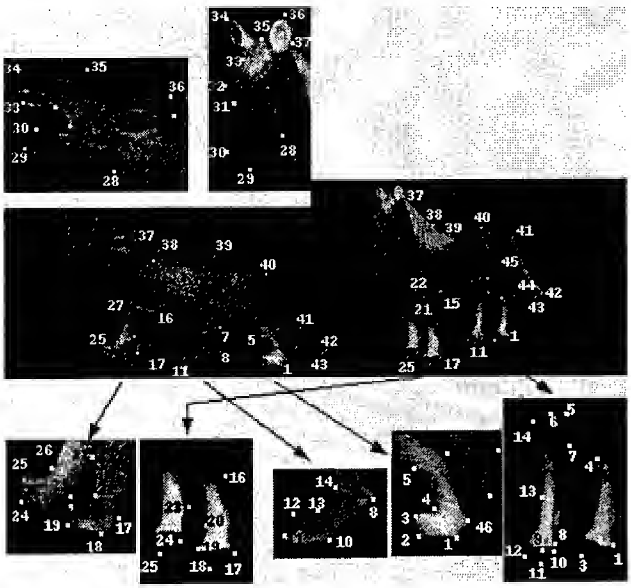


Figure 3: Qualitative match between different objects. Note the correct correspondence between the feet of the wolf to the feet of the horse, and the correspondence between the tails. Numbers appearing on the image of only one object denote features that were not matched (2,26,27 on the occluding contour of the wolf, and 9,46 on the horse).

of view (figure 4). The matching is successful if the two silhouettes are similar enough (as 2D entities). Note that preservation of shape under change of viewpoint is a quality that defines "canonical views". Using our similarity function we can learn these views from examples.

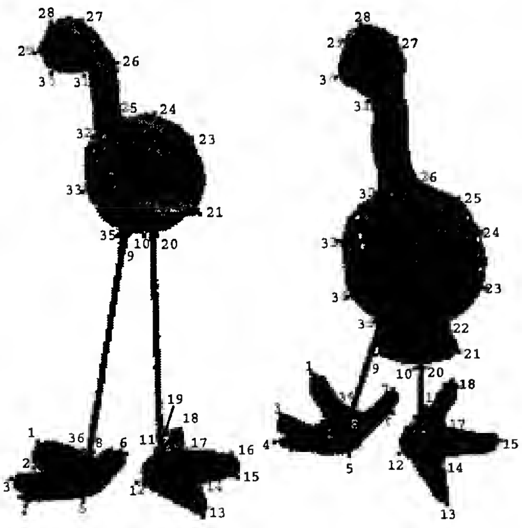


Figure 4: Feature correspondence between two different views of an object. Note the large forthshortening effect, that does not destroy local similarity. Segment merging is seen at point 16 of the left image, and points 7,22,34 of the right one.

We note that all the examples above (including the one from figure 1) were generated with the same values of parameters: $w_1 = 1$, $w_2 = 0.8$, $w_3 = 8.0$, $K = 4$; the parameters were not tuned to accomplish optimal performance. These values were obtained in an ad-hoc fashion, however, and not by systematic optimization (which is left for a future research). Nevertheless, we noticed that the matching results are stable, namely, the same results are obtained under quite large perturbations of the parameters' values.

3.2 Silhouette classification

Finally, we show an application of our matching method to silhouette classification. Successful classification is an indirect evidence for the correct matching obtained between a large number of image pairs (there were 4005 matching assignments involved in the study described below).

The task is defined as follows: 90 images of 6 different objects are given. The objects include toy models of a cow, wolf, hippopotamus, two different cars, and a child. Each object contributed 15 images, taken from different points of view (in a sector range of 40° azimuth and 20° elevation). Using automatic image segmentation techniques, the outlines of the objects were extracted from a black background, and automatically matched to each other. Based on the feature matching, a dissimilarity measure was computed for every pair of silhouettes. These distances served as input to an automatic hierarchical clustering algorithm [2], in order to divide the contours into subsets, where contours in the same subset are more similar to each other than contours in different subsets.

More specifically, we computed the 90×90 dissimilarity matrix for our database of 90 images. Since the matrix is symmetric and the diagonal elements vanish, we had to match 4005 image pairs, a task which took 6.5 hours on an INDY R4400 175Mhz workstation (5.8 seconds per match, on average).

The distance matrix was fed into a clustering algorithm [2], which produced the dendrogram of figure 5. In the final level of classification (the lowest level in the hierarchy), the 90 images were grouped precisely according to their identity. But even more interesting is the hierarchical structure which emerged. The scale parameter of the clustering algorithm (temperature) defines the level of specification, and it reflects our intuition regarding families of objects. This structure wouldn't have emerged if the estimation of the distance between weakly similar shapes was not reliable. Therefore, capturing the true hierarchical structure stands as an indirect evidence for the quality of our matching algorithm.

The clustering algorithm that we used does not require the embedding of our data points (images) in some normed space. The only information which is needed is pairwise distances. This is in contrast with most other clustering algorithms, including various kinds of iterative algorithms, that perform data partitioning through estimation of means (e.g., using deterministic annealing). The use of means involves two main assumptions: that the data points are vectors in some space, and that their distribution is central. The algorithm that we have used avoids these two assumptions.

On the other hand, for visualization purposes it might be useful to obtain a low dimensional representation of the images as points in a vector space. Such a representation is not always possible, and can be misleading. One popular way to obtain a low dimensional vector representation of proximity data, is by the method known as Multi Dimensional Scaling, which is a non linear optimization method. Here we adopt another approach to facilitate visualization: to each image i we assign a vector, whose k-th component is the distance between image i and image k. This defines an embedding of N images in R^N.

Next, we lower the dimension of the representation using Principal Component Analysis. For the set of 90 images described above, the results of these manipulation are shown in figure 6. For illustration purposes, the grouping of points was done manually, and the process was repeated for the two groups of animals and cars.

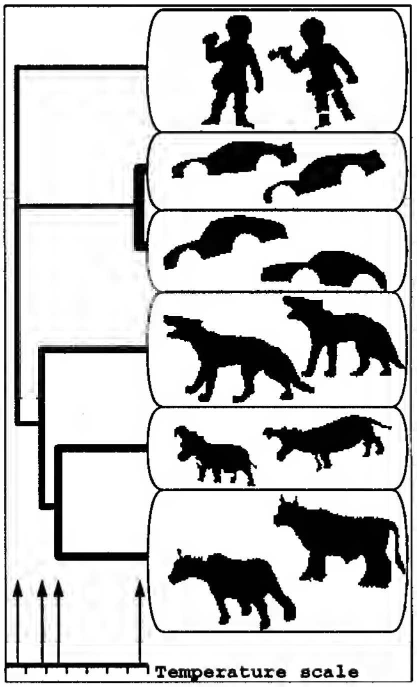


Figure 5: A classification tree (dendrogram) obtained by a hierarchical clustering algorithm, using the pairwise distances between silhouettes. Two arbitrary representatives are shown for each class. The automatic classification is 100% correct, and the hierarchy reflects the true structure in the database.

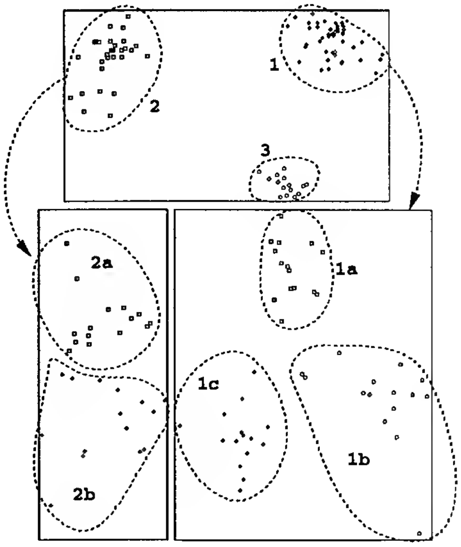


Figure 6: Visualization of the mutual similarities between the 90 images. At the highest level, groups 1,2,3 represent the images of animals, cars and child respectively. The groups which contain more then one object can be projected again onto their own principal directions. The grouping was done manually, for illustration only.

References

[1] N. Ansari, E. J. Delp, "Partial shape recognition: a landmark based approach", *PAMI*, vol 12, pp. 470-489, 1990.

[2] M. Blatt, S. Wiseman and E. Domany, "Super-paramagnetic clustering of data", *Physical Review Letters*, vol. 76, pp. 3251, 1996.

[3] D. Geiger, A. Gupta, L. A. Costa and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours", *PAMI*, vol 17, pp. 294-302, 1995.

[4] J. W. Gorman, O. R. Mitchell and F. P. Kuhl, "Partial shape recognition using Dynamic programming", *PAMI*, vol 10, pp. 257-266, 1988.

[5] M. W. Koch, R. L. Kashyap, "Using polygons to recognize and locate partially occluded objects", *PAMI*, vol 9, pp. 483-494, 1987.

[6] H. Liu, M. D. Srinath, "Partial shape classification using contour matching in distance transformation", *PAMI*, vol 12, pp. 1072-1079, 1990.

[7] N. Ueda, S. Suzuki, "Learning visual models from shape contours using multiscale convex/concave structure matching", *PAMI*, vol 15, pp. 337-352, 1993.

Appearance Based Object Recognition with Illumination Invariance

Kohtaro Ohba Yoichi Sato Katsusi Ikeuchi

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213

Abstract

This paper describes a method for recognizing partially occluded objects under different levels of illumination brightness by using the eigen-space analysis. In our previous work, we have developed the "eigen-window" method to recognize the partially occluded objects, and have demonstrated that the method works successfully for multiple objects with specularity under constant illumination.

In this paper, we modify the eigen-window method for recognizing objects under different illumination conditions by using additional color information. In the proposed method, a measured color in the RGB color space is transformed into the HSV color space. Then, the hue of the measured color, which is invariant to change in illumination brightness and direction, is used for recognizing multiple objects under different levels of illumination conditions.

The proposed method was applied to real images of multiple objects under different illumination conditions, and the objects were recognized and localized successfully.

1 Introduction

Object recognition has a wide variety of military and civilian applications. Some of the representative applications include bin-picking, automatic target recognition, and surveillance and monitoring. Some of the earlier work in this domain include [1] - [2]. Despite the long history of research, these applications still provide a challenge to computer vision researchers. The main difficulties include requirement for real-time processing, difficulty in segmentation, and difficulty in obtaining appropriate object models.

Recently, visual learning methods based on the eigen-space analysis have shown a potential to solve some of these difficulties [3] and [9]. In the eigen-space analysis, object models are *learned* from a series of images taken in the same environment as in the recognition mode. Thus, the difficulty of object modeling is avoided in the analysis. Furthermore, since an object model is stored as a vector in a low dimensional eigen-space, and since objects are recognized by comparing the model with image vectors, computation for object recognition can remain effectively low enough to achieve real-time performance.

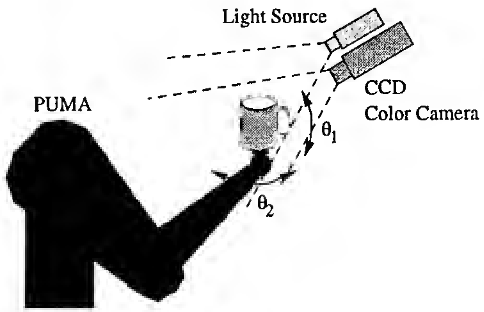


Figure 1: Experimental Setup

Although promising, the current eigen-space analysis is based on the assumption that objects are not occluded in images. Therefore, to apply the eigen-space analysis for partially occluded objects, we proposed to divide appearances into small windows, referred to as *eigen-windows* [13] and to apply eigen-space analysis to each eigen-window. The basic idea is that, even if some of the windows are occluded, a large number of windows is still visible that the object can be recognized and localized in the images.

One drawback of the eigen-window method is that only a limited number of images can be used for learning object models, and therefore, all possible illumination directions cannot be taken into account. Therefore, the object may be illuminated from a different direction at the recognition mode, resulting in incorrect recognition results.

In this paper, to overcome that drawback, we propose to use the color measurement *hue*, which is illumination invariant, in the eigen-window method. To demonstrate the effectiveness of the proposed method, we applied the method to real images taken under different illumination directions and brightness.

2 Eigen-Window Method

In this section, we briefly review the eigen-window method that we have proposed [13] to overcome limitations of the original eigen-space analysis, such as image shift, occlusion, noise, and scaling.

2.1 Eigen-Space Technique

Let M be the number of the images $z_1, z_2, \cdots, z_M$ in a training set related to each rotation of view points θ_1 and θ_2, as shown in Figure 1. Each image z_i, with dimensions $N \times N$, has been converted into a column vector of length N^2.

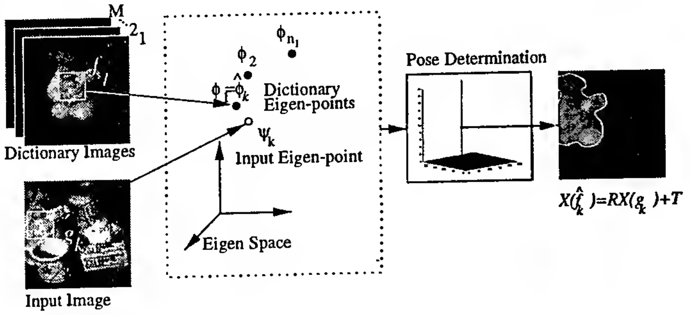


Figure 2: Eigen-Window Technique.

By subtracting the average brightness c of all the images, we obtain the training matrix of the size of N^2 by M,

$$Z = [z_1 - c, z_2 - c, \cdots, z_M - c]. \quad (1)$$

This covariance matrix $Q = ZZ^T$ provides a series of eigenvalues λ_i and eigenvectors $e_i (i = 1, \cdots, N^2)$, where each corresponding eigenvalue and eigenvector pair satisfies:

$$\lambda_i e_i = Q e_i. \quad (2)$$

For reducing memory requirement, we ignore eigenvectors corresponding to small eigenvalues $e_i (i > l)$. These eigenvectors do not affect object recognition results significantly. Once we obtain the remaining eigenvectors, we can construct the eigenvector matrix $E = [e_1, e_2, \cdots, e_l]$ which projects an image z_i (dimension N^2) into the eigen-space as an *eigen-point* g_i (dimension l).

$$g_i = E^T (z_i - c). \quad (3)$$

The eigen-space analysis can drastically reduce the dimension of the images (N^2) to the eigen-space dimension (l) while preserving enough dominant features to reconstruct the original images.

2.2 Eigen-Window Technique

To reduce the disturbance effects such as image shift and occlusion, we propose to select small windows in the original images. Each of the selected small windows is then analyzed by using the eigen-space analysis as described in the previous section. We call this method *the eigen-window method.* Figure 2 shows the overview of the method.

2.2.1 Training Eigen-Windows

The training set of eigen-windows is given as:

$$F = [F^1, F^2, \cdots, F^M], \quad (4)$$

where F^i denotes the collection of eigen-windows from the ith training image. Each F^i has the form $[f_1 - c, f_2 - c, \cdots, f_{n_i} - c]$, where f_j denotes the jth eigen window in the ith training image; n_i denotes the number of eigen-windows in the ith image; and c is the average intensity value across all eigen-windows in the whole training set. In Figure 2, the white square denotes one of the training eigen-windows.

2.2.2 Matching Operation

From an input image, a set of input eigen-window images is obtained:

$$G = [g_1 - c, g_2 - c, \cdots, g_n - c], \quad (5)$$

such as the white window in the lower left image in Figure 2.

The similarity between a training eigen-window and an input eigen-window is evaluated by using the distance between them in the eigen-space. Given an input eigen-point ψ_k projected from input eigen-window g_k using equation (3), we try to find a training eigen-point $\hat{\phi}_k$ from all training eigen-points ϕ projected from all training eigen-windows f. The training eigen-point is the one with the maximum similarity defined as:

$$\hat{\phi}_k = \arg \min_{\forall \phi} (||\psi_k - \phi||), \quad (6)$$

where $||x||$ denotes the norm of x using L1-norm or L2-norm. We denote the eigen-window that is projected to $\hat{\phi}_k$ as $\hat{f}_k$. The eigen-window $\hat{f}_k$ corresponds to the input eigen-window g_k.

2.2.3 Voting Operation

The previous matching operation selects a set of training eigen-windows, $[\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n]$ corresponding to input eigen-windows. We now sort the selected training eigen-windows into each groups, which contains windows that comes from the same training images, from a corresponding training image:

$$\left[\hat{F}^1, \hat{F}^2, \cdots, \hat{F}^M\right], \quad (7)$$

where

$$\hat{F}^i = \{\hat{f} | \hat{f} \; comes \; from \; training \; image \; i\} \quad (8)$$

We then prepare a pose space for voting from the established correspondences. In this operation, we consider only translation, and therefore the space is two dimensional corresponding to a $[x, y]$ location of eigen-windows. The size of the pose space is set to be twice that of the input image size, e.g., 256 × 256 in our examples. The pose space is prepared separately for each group $\hat{F}^i$.

Using each correspondence, we can compute the difference of the training eigen-window's location $X(\hat{f}_k)$ and the input eigen-window's location $X(g_k)$. ($X(f)$ represents $[x, y]$ location of the eigen-window f.) The difference is given as $X(g_k) - X(\hat{f}_k)$.

Then, in the pose-space, the cell that represents this distance, $X(g_k) - X(\hat{f}_k)$, gets a vote. To avoid the digitization error, all of the 5×5 neighbor cells around the center cell get a vote from a single correspondence. We repeat this operation, using all correspondences for a group $\hat{F}^i$ (all the correspondences from the same training image.) Then, we obtain a resulting pose-space for each of the groups $\hat{F}^i$.

Some small peaks in the pose-space are due to noise; other prominent peaks are due to actual objects in an input image. By thresholding these peaks, we can eliminate noise peaks and extract only prominent peaks.

2.2.4 Pose Determination

The number of the prominent peaks in the pose-space is equal to the number of objects that have roughly the same rotation, but have a different translation.

By retrieving voted pairs, we further divide the group $\hat{F}^i$ into sub-groups, each of which belongs to each prominent peak, i.e., an isolated object in the input image.

Since the training set is sampled along the rotation dimension, there exists a side effect of small object rotation due to the finite sampling interval. To obtain the rotation and the translation precisely, we refine the pose estimate via a least square minimization, using the pairs in each sub-group.

$$X(\hat{f}_k) = RX(g_k) + T, \tag{9}$$

where R and T denote the small rotation and translation, respectively.

2.3 Selection of Effective Eigen-Windows

In the eigen-window method, it is very important and also difficult to select an optimal set of training eigen-windows. If all of initially selected windows are used as the eigen-windows, two problems occur: 1) the number of eigen-windows becomes very large and storing them requires a large amount of memory, and 2) due to the similarity among eigen-windows, the matching process becomes erroneous.

In this section, we introduce three criteria for selecting the optimal set of eigen-windows: *detectability*, *uniqueness*, and *reliability*.

The detectability measures the ease of detecting a window in an entire image. For example, a window containing corners of an object is much easier to detect than one containing a planar region.

Although some windows are easy to detect, they may be similar to each other. This situation happens when the target object has multiple similar corners. To select truly, distinct windows, we introduce a global goodness measure called the uniqueness measure.

In addition, the reliability measure is used to select windows which do not appear and disappear with small variation of object pose such as orientation and translation.

Using these three measures, we can obtain the optimal set of eigen-windows.

2.3.1 Detectability: Local Goodness

For initial selection of eigen-windows in images, algorithms to select feature points for object tracking can be used. Tomasi *et.al.* proposed to use the following 2×2 matrix as the trackability measure for a window $X = [x, y]^T$ [12].

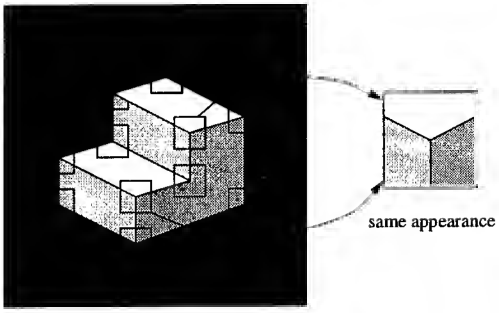


Figure 3: A Problem in Corner Detection with Trackability.

$$G = \sum_{X \in R} (\frac{\partial I}{\partial X})(\frac{\partial I}{\partial X})^T. \tag{10}$$

where I represents pixel intensities inside the window.

This matrix G has two eigenvalues λ_1 and λ_2. The window is accepted as a good one, if the equation

$$min(\lambda_1, \lambda_2) > \lambda. \tag{11}$$

holds, where λ is a predefined threshold. The measure works well for detecting most of the important corners.

2.3.2 Uniqueness: Global Goodness

By using the detectability measure, we can select windows which contain important features. Unfortunately, however, the detectability measure does not guarantee the global uniqueness of the selected windows. In other words, some of the selected window may have corners very similar to corners in other windows (Figure 3). These windows are not desirable for recognition and localization of objects.

We define the global goodness of windows as the uniqueness of an eigen-window. The uniqueness of each window can be measured by computing similarity among eigen-windows. As discussed in Section 2.2.2, the similarity between a training eigen-window and an eigen-window in an input image is computed by using the distance between them in the eigen-space.

We can use the same measure for evaluating the global goodness of selected windows, i.e., the similarity among training eigen-windows.

$$S_{l,m} = ||\phi_l - \phi_m|| \leq T_{sim}. \tag{12}$$

The similarity $S_{l,m}$ between two training eigen-points ϕ_l and ϕ_m, which are projected from two eigen-windows, is evaluated by using the equation. If the computed similarity is less than a certain threshold T_{sim}, then the two eigen-windows are discarded from the training set (Figure 4).

The elimination of similar eigen-windows reduces the size of a training set effectively. Also, the elimination makes the matching process more robust. That is because the reduced set does not contain similar eigen-windows, and therefore the matching evaluation is less likely to be fooled by similar eigen-windows.

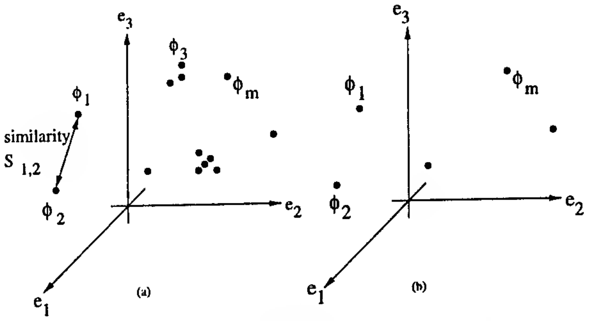


Figure 4: Eigen-space Distribution in 3D

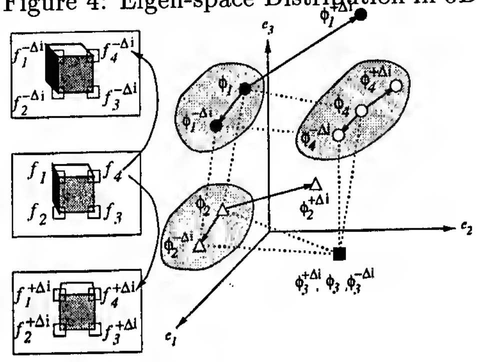


Figure 5: Reliability Evaluation in Eigen-space

2.3.3 Reliability

The reliability of an eigen-window for object recognition can be inferred by measuring how much the eigen-point which corresponds to the eigen-window moves in the eigen-space while an object is moved slightly.

For example, if training images are taken by rotating the object at the angle step $10deg$, we can evaluate the reliability by rotating the object slightly, e.g., $\pm 5deg$, for each of the training images. If an eigen-point ϕ_j remains close to the original location for the training image, we consider the eigen-point to have high reliability. The reliability $R_j^{\pm\Delta i}$ is defined as:

$$R_j^{\pm\Delta i} = \sum_{\delta i \in \pm\Delta i} \|\phi_j^{\delta i} - \phi_j\| \leq T_{rel}, \qquad (13)$$

where T_{rel} is a threshold value for the reliability (Figure 5).

Some of the windows may disappear and reappear while the object is rotated. In this case, we simply discard those feature points.

3 Illumination Invariance

In our previous work [13], we have shown that the eigen-window method can successfully recognize and localize an object in b/w input images which contain multiple objects with specularity, even if the input images contain significant amount of noise, occlusion, image shifting and scaling change.

However, the method was based on an assumption that the location and brightness of a light source are fixed. Therefore, the method did not take into accounts shading variation such as highlights on object surfaces. For instance, if an object exhibits specularity, the object appearance can change drastically with different illumination directions, which confuses recognition and localization of the object.

To overcome this limitation, we propose to use an illumination invariant measure for the eigen-window method. By using the illumination invariant measure, the eigen-window method can be used successfully for recognizing and localizing multiple objects under different illumination conditions.

3.1 Illumination Invariance: Hue

Instead of black-and-white intensity images, we use RGB color images in the modified eigen-window method. Actually, several pieces of research works were done on the color indexing in the past [17] - [19], but we would like to use the *hue* criterion for its simpleness, and a color image measured in the RGB color space is converted to a HSV image (H: Hue, S: Saturation, V: Value). In these three parameters, the hue parameter is the value which represents color information, e.g., without brightness. Therefore, the hue is not affected by change of the illumination brightness and direction if the following two conditions hold: 1) the light source color can be expected to be almost white, and 2) a saturation value of object color is sufficiently large.

The original color of object X is transferred to be $X' = s \cdot X + t \cdot I$ by the change in diffuse shading and specularity as shown in Figure 6. s and t represent a relative strength of the diffuse reflection component and the specular reflection component of the color X', respectively. If the two conditions mentioned above are true, then the hue of X' remains the same as that of X.

In Figure 6, object color is represented by three color components S_1, S_2, and S_3. In the RGB color space, those three color components are Red, Green and Blue. Then, the light source color I is given as $I = (1, 1, 1)$. To define hue, saturation and intensity, one pair from three components, Red, Green, and Blue, have to be assigned to S'_1 and S'_2. Usually, Red and Green are assigned as $S'_1 = R$ and $S'_2 = G$.

We conducted a simple experiment using a color test chart to see how hue is affected under different levels of illumination brightnesses. The result is shown in Figure 7 and Figure 8. In Figure 8 (b), we can see that hue remains almost constant over a wide range of illumination brightness for many color blocks.

However, for some color blocks, the value of hue does change with different levels of illumination brightnesses. For instance, the black-white color blocks in the last row of the color chart (color blocks #30-#35), red (color block #12) and magenta (color block #24).

That is because the saturation of color blocks #30-#35 is not sufficiently large, i.e., they are very close to gray. Also, hue has a discontinuity at 0 and 2π. That is the reason for unstable hue of the color blocks #12 and #24.

To obtain the value of hue reliably, we propose to use three criteria: *intensity value*, *saturation*, and *phase*.

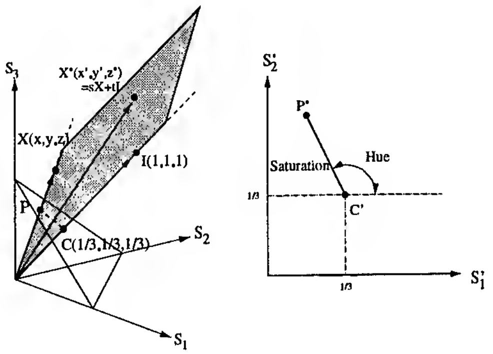


Figure 6: HSV Space.

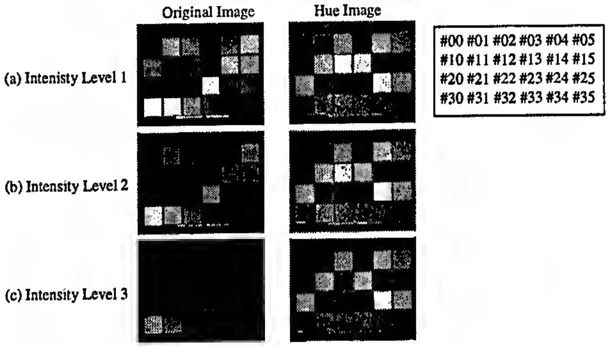


Figure 7: Illumination Constancy with a Color Test Pattern Image.

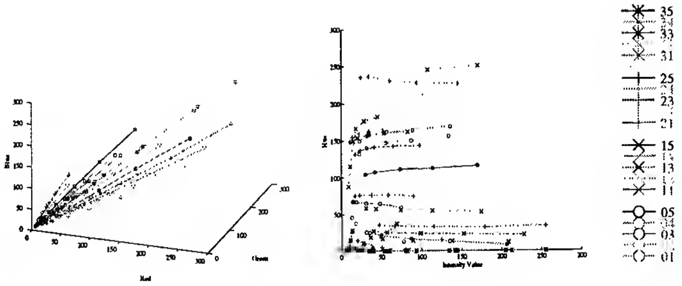


Figure 8: Color Elements. (a) RGB space; (b) Hue-Intensity space

3.1.1 Intensity Value

To eliminate the background noise, we apply a threshold value for the intensity value as

$$if\ V < V_t\ then\ H = 0, \tag{14}$$

where V, V_t, and H are an intensity value, the threshold value, and a hue value, respectively. If measured color is not bright enough, the color is discarded. Then, the hue value is set to a predetermined value, i.e., 0.

3.1.2 Saturation

One of the problems shown in the example in Section 3.1 is that, if object color is close to gray, then hue value of the color is not stable. The reason is that, if the color is almost gray, the object color in $S'_1S'_2$ plane exists around the point C' in Figure 6. That means the hue angle cannot be determined robustly in the face of image noise. Therefore, measured color should be discarded if the saturation value is less than a certain threshold S_t:

$$if\ S < S_t\ then\ H = 0, \tag{15}$$

where S is the saturation value. Using the equation, measured color close to gray is discarded in the image.

3.1.3 Phase

The other problem shown in the example in Section 3.1 is that color close to red has a hue value near its discontinuity. The range of hue value is from 0 to 2π, and it has discontinuity at 0 and 2π. We avoid the discontinuity effect by using the phase threshold value ΔP_t as:

$$if\ H < \Delta P_t\ or\ \|H - 2\pi\| < \Delta P_t\ then\ H = 0. \tag{16}$$

In the examples shown in Figure 7 and Figure 8, the red color element may be neglected with this criterion.

It is important that the discontinuity of hue value depends on the selection of the color components S'_1 and S'_2. In the next section, we discuss how to select the color components S'_1 and S'_2 to be able to find more windows.

3.2 How to Select Color Components S'_1 and S'_2

Usually, the two color components S'_1 and S'_2 are set to R and G. But if the R and G factors are used for the two color components, the discontinuity of hue appears around the color red as described in the previous section. Therefore, if red is the most important component for recognizing the objects, the use of R and G for S'_1 and S'_2 is not desirable.

In this section, we show how to choose the S'_1 and S'_2 from RGB components so that we can select more windows to be used as eigen-windows as described in Section 2.

There are six combinations for the selection of S'_1 and S'_2 from the RGB components. Figure 9 shows the

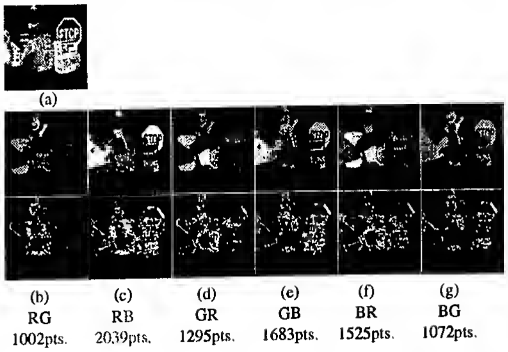


Figure 9: Image Invariance and Window Selection. (a) Original Image; (b) Hue Image and Feature Points with RG; (c) RB; (d) GR; (e) GB; (f) BR; (g) BG

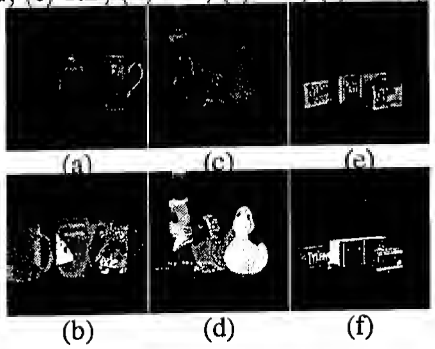


Figure 10: Some Example Hue Images. (a) Original Image of Mug; (b) Hue Images of Mug; (c),(d) Birds; (e),(f) Tylenol

result of window selection by using each combination of S_1' and S_2'. In the figure, RG represents that $S_1' = R$ and $S_2' = G$.

The windows in the hue images were selected by using the corner detector algorithm as described in Section 2.3.2. So if a hue image does not have enough contrast, fewer windows are selected. In this example, the largest number of windows was selected for the case of $S_1' = R$ and $S_2' = B$ in Figure 9. Intuitively, that result indicates that there are not many green color components in the example image.

Several examples of hue images are shown in Figure 10. In those examples, we can see that the hue value remains almost constant on the object surface with large shading change. For instance, in Figure 10 (c) and (d), the hue of the yellow duck's surface appears to be constant even though the color image of the duck has a wide range of intensity. In those examples, background, gray color, and green color have been eliminated according to the equations (14), (15) and (16), respectively.

4 Experimental Results

The proposed method was used for recognition and localization of objects in three test cases. In the first case, the same illumination condition was used both for training and for input images. In the second case, input images were taken under different levels of illumination brightnesses. In the last case, input images were taken with different light source locations.

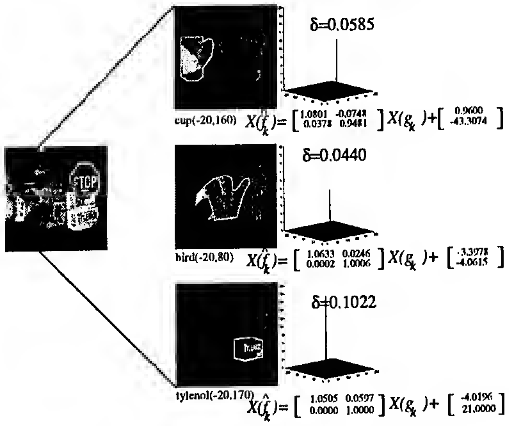


Figure 11: Recognition Result

4.1 Object Recognition and Localization with Hue Image

First, a set of training eigen-windows was obtained as described in Section 2. The training images were taken at $\theta_1 = [-20, 0, 20]$ and $\theta_2 = [0, 10, 20, \cdots, 350]$ for three different objects, *mug*, *bird*, and *tylenol*. We refer to the original images as $type(\theta_1, \theta_2)$. For example, the image $mug(-20, 60)$ denotes the image for the mug taken at the position $\theta_1 = -20deg$ and $\theta_2 = 60deg$. One hundred eight images were taken for each of the objects by using the experimental setup shown in Figure 1.

Then, eigen-windows were selected in each training image by using the detectability, similarity and reliability measurements as described in Section 2.3. The number of eigen-windows for each of the objects was initially more than 8,000. After the three measurements were applied, less than 2,000 of the training eigen-windows were finally obtained. Then, these eigen-windows were projected to produce eigen-points according to the equation (3).

One input image containing multiple objects was taken as shown in left hand side of Figure 11. In the input image, there are 7 objects, *duck*, *mug*, *barney*, *bird*, *stop-sign*, *tylenol*, and *tylenol-cold*. First, eigen-windows were selected in the input image by using the detectability measure. Then, we established correspondences between the input eigen-windows and the training eigen-windows by using the similarity between their eigen-points according to the equation (12).

The recognition and localization results are shown in figures in the middle column in Figure 11. The figures in the right column show the resulting pose spaces. Also, the obtained affine parameters and standard deviations in the pose space are shown. As we can see, each object's type, pose, and location were successfully obtained.

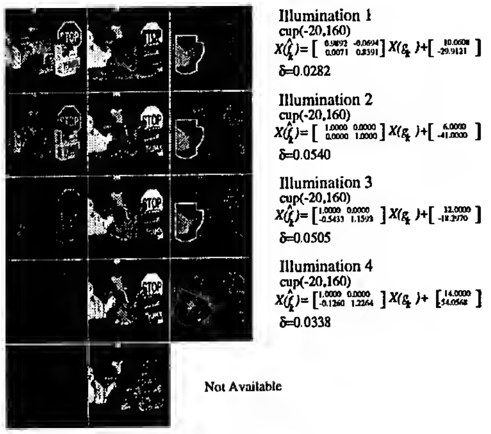

Illumination 1
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}0.9892 & -0.0694\\ 0.0071 & 0.8391\end{bmatrix}X(g_k)+\begin{bmatrix}10.0600\\ -29.9121\end{bmatrix}$
δ=0.0282
Illumination 2
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.0000 & 0.0000\\ 0.0000 & 1.0000\end{bmatrix}X(g_k)+\begin{bmatrix}6.0000\\ -41.0000\end{bmatrix}$
δ=0.0540
Illumination 3
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.0000 & 0.0000\\ -0.5433 & 1.1593\end{bmatrix}X(g_k)+\begin{bmatrix}12.0000\\ -18.2970\end{bmatrix}$
δ=0.0505
Illumination 4
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.0000 & 0.0000\\ -0.1260 & 1.2264\end{bmatrix}X(g_k)+\begin{bmatrix}14.0000\\ 54.0568\end{bmatrix}$
δ=0.0338
Not Available

Figure 12: Object Recognition Results with Illumination Change

4.2 Effect of Illumination Brightness Change

The same training eigen-window set was applied to input images taken under a wide range of illumination brightness. Figure 12 shows the result.

The original color images are shown in the left column, and the computed hue images are shown in the middle column. The localization and recognition results are shown in the right column. The affine parameters and standard deviations of the pose space are also given in the figure.

The hue images did not change significantly with different levels of illumination brightness. The main difference between the hue image for the brightest illumination and that for the darkest illumination is that hue values were not computed over a large portion of the object surfaces. This is because intensity values were so small that hue values were set to zero as the background value according to the equation (15).

The experimental results show that the proposed method works even when input images are taken with different levels of illumination brightness. The object was recognized and localized successfully.

4.3 Effect of Different Light Source Positions

The proposed method was also applied to input images taken with different light source locations. As the light source position changes, the appearance of objects in input images changes drastically. Therefore, changing light source position makes recognition and localization of objects even harder than changing illumination brightness.

In this experiment, four different light source positions were used as shown in Figure 13. The left column images of Figure 14 show the input images taken with each of the four light source positions. The middle column images of Figure 14 show the obtained hue images. The right column images present the recognition and localization results. The affine parameters and standard deviations of pose-space are also shown in the figure.

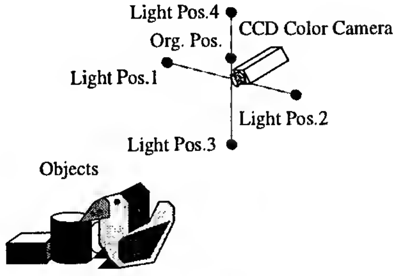


Figure 13: Light Source Position

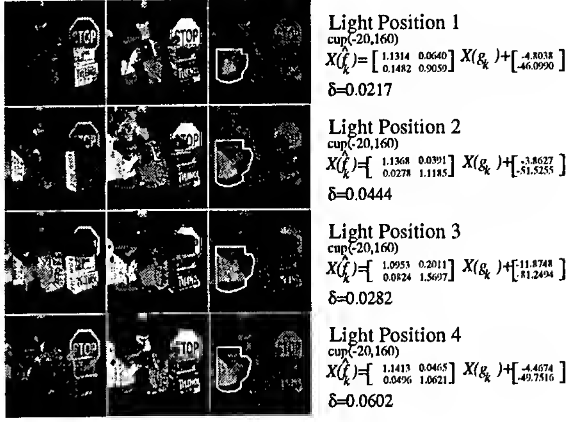

Light Position 1
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.1314 & 0.0640\\ 0.1482 & 0.9059\end{bmatrix}X(g_k)+\begin{bmatrix}-4.8038\\ -46.0990\end{bmatrix}$
δ=0.0217
Light Position 2
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.1368 & 0.0391\\ 0.0278 & 1.1185\end{bmatrix}X(g_k)+\begin{bmatrix}-3.8627\\ -51.5255\end{bmatrix}$
δ=0.0444
Light Position 3
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.0953 & 0.2011\\ 0.0824 & 1.5697\end{bmatrix}X(g_k)+\begin{bmatrix}-11.8748\\ -81.2494\end{bmatrix}$
δ=0.0282
Light Position 4
cup(-20,160)
$X(\hat{f}_k)=\begin{bmatrix}1.1413 & 0.0465\\ 0.0496 & 1.0621\end{bmatrix}X(g_k)+\begin{bmatrix}-4.4674\\ -49.7516\end{bmatrix}$
δ=0.0602

Figure 14: Effect of Light Source Direction

Note that, in this experiment, there was no ambient illumination. Hence, the appearance of the objects change significantly with different light source positions. Nevertheless, the mug was correctly recognized and localized except in the input image for the light position 1. In this case, hue values were not obtained over a large portion of the object surface because of shadow casting on the surface.

5 Conclusion

In this paper, we described a novel method called the eigen-window method. The method extends the standard eigen-space analysis for the case of recognizing partially occluded objects. To reduce the redundancy among eigen-windows, we proposed three measures for selecting eigen-windows effectively: detectability, uniqueness, and reliability.

By using hue, which is an illumination invariant measure, the eigen-window method was extended further for recognition and localization of objects in images taken under changing illumination conditions. To use hue information of input images reliably, we introduced three criteria for computing hue values: intensity value, saturation, and phase.

The proposed method was applied to real images, and the method recognized and localized objects successfully even in images taken under significantly different illumination conditions.

Acknowledgements

This research has been supported in part by the Advanced Research Projects Agency under the Department of the Army, Army Research Office grant number DAAH04-94-G-0006. Views and conclusions con-

tained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army or the United States Government.

References

[1] Y. Fukada, H. Doi, K. Nagamine, and T. lnari, "Relationships-Based Recognition of Structural Industrial Parts Stacked in Bin," *Robotica*, Vol.2, pp.147-154, 1984.

[2] B. K. P. Horn and K. lkeuchi, "The Mechanical Manipulation of Randomly Oriented Parts," *Scientific American*, Vol.251, No.2, pp.100-111, 1984.

[3] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," Proc. CVPR 1991, pp.586-591, 1991.

[4] M. A. Turk and A. P. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Newroscience*, Vol.3, No.1, pp.71-86, 1991.

[5] S. Sclaroff and A. P. Pentland, "A Model Framework for Correspondence and Description," *Proc. 4th Int'l Conf. on Computer Vision*, pp.308-313, 1993.

[6] A. P. Pentland and B. Horowitz, "Recovery of Nonrigid Motion and Structure," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.13, No.7, pp.730-742, 1991.

[7] A. P. Pentland, B. Moghaddam, T.Starner, "View-Based and Modular Eigenspaces for Face Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[8] B. Moghaddam and A. P. Pentland, "Probabilistic Visual Learning for Object Detection," *The 5th International Conference on Computer Vision*, 1995.

[9] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, Vol.14, No.1, pp.5-24, 1995.

[10] M. Uenohara and T. Kanade, "Vision-Based Object Registration for Real-Time lmage Overlay," *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, Nice France, April 1995.

[11] C. Tomasi and T. Kanade, "Shape and Motion from lmage Streams: a Factorization Method-2. Point Features in 3D Motion," *Technical Report*, CMU-CS-91-105, Carnegie Mellon University, Pittsburgh, PA January 1991.

[12] C. Tomasi and T. Kanade, "Shape and Motion without depth," *Proc. of the Third International Conference in Computer Vision*, Osaka, Japan, December 1990.

[13] K. Ohba and K. lkeuchi, "Recognition of the Multi Specularity Objects using the Eigen-Window," *Proceeding of International Conference on Pattern Recognition*, August 1996.

[14] J. Krumm, "Eigenfeatures for Planar Pose Measurement of Partially Occluded Objects," *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 1996.

[15] R. Bajcsy, S. W. Lee and A. Leonardis, "Detection of Diffuse and Specular lnterface Reflections and lnter-Reflections by Color lmage Segmentation," *International Journal of Computer Vision*, Vol.17, No.3, pp.241-272, 1996.

[16] S. K. Nayar and R. M. Bolle, "Reflectance Based Object Recognition," *International Journal of Computer Vision*, Vol.17, No.3, pp.219-240, 1996.

[17] G. Healey and D. Slater, "Global color constancy: recognition of objects by use of illumination-invariant properties of color distribution," *Journal of Optical Society of America*, Vol.11, No.11, pp.3003-3010, Nov. 1994.

[18] M. J. Swain, "Color lndexing," *International Journal of Computer Vision*, Vol.7, No. 1, pp.11-32, 1991.

[19] B. V. Funt and G. D. Finlayson, "Color Constant Color Indexing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.17, No.5, pp.522-529, May 1995.

[20] D. Slater and G. Healey, "The lllumination-lnvariant Recognition of 3D Objects Using Local Color lnvariants," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.18, No.2, pp.206-210, Feb. 1996.

[21] C. L. Novak and S. A. Shafer, "Supervised Color Constancy Using a Clor Chart," *Technical Report*, CMU-CS-90-140, School of Computer Science, Carnegie Mellon University Pittsburgh PA, June 1990.

[22] D. Aubert and C.Thorpe, "Color lmage Processing for Navigation: Two Road Trackers," *Technical Report*, CMU-Rl-TR-90-09, The Robotics lnstitute, Carnegie Mellon University Pittsburgh PA, April 1990.

Generic, Model-Based Edge Estimation in the Image Surface

Pei-Chun Chiang, Thomas O. Binford

Robotics Laboratory, Computer Science Department
Stanford University, Stanford CA 94305

Abstract

Edge detection is the process of estimating extended curve discontinuities in the image intensity surface. As a primary module in image understanding systems, its performance is critical. This paper presents a new family of algorithms for detecting edges in an image. Edges with delta, step and crease cross sections along curves are found. In the algorithm, an image is convolved at each pixel at four orientations. Then, the best least-squares estimate of an edge is made over a 2x2x2 cube determined by adjacent x, y, and φ values. Finally, the detected edge image is built by a primitive linking process. Both theoretical analyses and simulation experiments have been done; real images have been analyzed. Quantitative results indicate that the proposed detector has superior detection and localization; the standard deviations of orientation and transverse position are a factor of four better than the Wang-Binford operator at the same signal to noise ratio. It performs well even when the ratio is less than 4.

1 Introduction

In the Successor paradigm, objects are interpreted globally by Bayesian networks that express probabilities among a hierarchy of uncertain geometric hypotheses. The 3D levels in Successor include: object (physical), 3D volume, 3D surface, 3D curve, and 3D point. Quasi-invariants and invariants correspond 3D surfaces to 2D image areas, 3D curves to 2D image curves, and 3D points to 2D image points. Successor requires reasonable measurement of extended image edges and of relations among extended edges in order to build 3D interpretations.

The performance of the edge detector affects most modules in image understanding systems, like stereo, motion, and surface estimation. Until recently, the quality of measurement of extended edges extracted from images was very unsatisfactory from the standpoint of vision systems. The Wang-Binford operator provided reasonable extended edges. This algorithm provides substantial improvements in the quality of measurement of extended edges.

An intensity image $I(x,y)$ is a mathematical surface. Edges in an image correspond to intensity discontinuities in the image surface. Low-order discontinuities are interesting, i.e. discontinuities along curves with cross sections in the form of delta, step, and crease, as shown in Fig. 1 [1].

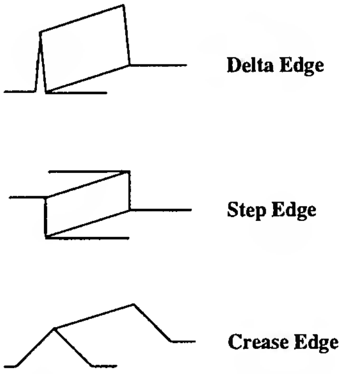


Figure 1: The transverse profiles of the three main types of edges.

In the past, a variety of segmentation algorithms have been developed, like the Binford-Horn operator [2], the Marr-Hildreth operator [3], the Canny operator [4], the Nalwa-Binford operator [5], However, the results are not satisfactory, primarily because they don't produce extended edges with sufficient quality, thus, restricting their applications. Since edge linking to form extended edges is an exponential branching process, accuracy of edgel estimation was required to cut the branching to tolerable limits. With the Wang-Binford operator [6] [7], high quality extended edges have been obtained. Now, the operator described here performs substantially better.

There are three principal problems in previous edge detection algorithms, excluding the Wang-Binford operator. A main problem is that the shading effect was not taken into account; that is, the intensity of an image along an edge was modeled as a constant, which is not true in the real scene. The second problem is that the estimates of orientation and position were not accurate. The inaccuracy raised severe problems in linking pixels into extended edges, and then in the follow-up processes. The third problem is that only step edges were covered in those algorithms. There are three main types of edges: delta edges, step edges, and crease edges. A significant amount of information in an image is lost if only step edges can be detected.

A new edge detector which overcomes all these problems is presented here. It provides least-squares estimates of the position, orientation, and amplitude of an edge, which are no longer sensitive to shading. In addition, it estimates three kinds of edges by using a single family of algorithms. Though the Wang-Binford operator can also estimate all three edge types reasonably well, the performance of this detector improves dramatically over Wang-Binford from both practical and statistical standpoints. Based on accurate estimates and the statistical analyses of edgels, isolated edgels are aggregated into extended edges. The linking process is still under development, but the current results look promising. In this paper, a generic edge model is introduced in Section 2, then the edge detector is developed in Section 3. The statistical analysis is discussed in Section 4. Real images are tested in Section 5. The conclusions are given in Section 6.

2 Edge Model

2.1 Ideal Model

Edges are defined as discontinuities of the image intensity. According to the order of discontinuity, edges can be divided into three groups (see Fig. 1): delta and step edges with zeroth order discontinuity, and crease edges with first order discontinuity.

An extended edge in an image can be approximated locally by a tangent straight line element called an *edgel*. Near an edgel, the image surface can be expressed as $I_0(l,t)$, where l and t are the longitudinal and transverse directions of the edgel respectively. In general, $I_0(l,t)$ can be decomposed into two profile factors, $I_L(l)$ and $I_T(t)$:

$$I_0(l,t) = I_L(l) \times I_T(t)$$

where $I_L(l)$ is the image profile in the longitudinal direction and $I_T(t)$ is the image profile in the transverse direction.

In the longitudinal direction, the shading effect is considered, i.e.

$$I_L(l) = h + s_1\ l$$

where h is the amplitude of the edgel, and s_1 is the shading coefficient. Typically, the gray level of $|h|$ is larger than 2 and $|s_1|$ is much smaller than 0.1. The shading effect is included only in $I_L(l)$, not in $I_T(t)$, because the edgel profile dominates the intensity of an image in the transverse direction.

Different functions are used to model different types of edgels in the transverse direction. For delta edgels, $I_T(t)$ can be described by a delta function; for step edgels, $I_T(t)$ can be described by a step function; for crease edgels, $I_T(t)$ can be described by a triangle function. The three edgels are closely related: the derivative of a triangle function is a step function, and the derivative of a step function is a delta function. Hence, step edgels can be determined by convolving the image with the first derivative of the mask. Similarly, crease edgels can be determined by convolving the image with the second derivative of the mask. Most of the discussions in this paper will focus on detecting the delta edgel, but they can be applied to the other two types of edgels as well.

2.2 Real Model

A real image is blurred by the impulse response of the optical system, and is perturbed by measurement noise. Blur can be represented by convolving the original image with a 2-D blur function. In many cases, the blur function can be approximated by an isotropic 2-D Gaussian:

$$G(l,t) = \frac{1}{2\pi\sigma_b^2}\ \exp(-\frac{l^2+t^2}{2\sigma_b^2})$$

Noise can be modeled by additive, zero-mean, white Gaussian noise for which the probability density function $f_N(n)$ is

$$f_N(n) = \frac{1}{\sqrt{2\pi\sigma_N^2}}\ \exp(-\frac{n^2}{2\sigma_N^2})$$

Combining these two effects, the model of the delta edgel can be modified as

$$\begin{aligned} I(l,t) &= (I_L(l) \times I_T(t)) \otimes G(l,t) + N(l,t) \\ &= [(h + s_1\ l)\ \delta(t)] \otimes G(l,t) + N(l,t) \\ &\approx (h + s_1\ l)\ \exp(-\frac{t^2}{2\sigma_b^2}) + N(l,t) \end{aligned}$$

where $\otimes$ denotes 2-D convolution.

This is the generic edge model which will be used throughout the rest of the paper.

3 Edgel Detection

Based on the model developed in Section 2, a new edge detector will be presented in this section. The main function of this detector is to find the position (x,y), orientation α, and amplitude h of all edgels in an image. Parameters (x,y,α,h) will be estimated from the moments of an image, which are the convolutions of the image with a set of masks. Once these parameters are solved correctly, the edgels are detected.

3.1 Estimating the Moments

To detect the position, orientation, and amplitude of a delta edgel (also called a ridge), a mask M is designed as

$$\begin{aligned} & M(u,v) \\ =\ & G_U(u)G'_V(v) \\ =\ & \left[\frac{1}{\sqrt{2\pi\sigma_u^2}}\exp(-\frac{u^2}{2\sigma_u^2})\right]\left[-\frac{v}{\sigma_v^2}\frac{1}{\sqrt{2\pi\sigma_v^2}}\exp(-\frac{v^2}{2\sigma_v^2})\right] \end{aligned}$$

where (u, v) are the local coordinates of the mask. The profile of M along the v direction is the first derivative of a Gaussian, which will smooth the input image by convolving it with a Gaussian, and then estimate the directional derivative of the smoothed image. The profile of M along the u direction is another Gaussian, which will average the estimations of the derivatives.

Let moment k be the convolution of the image with the mask M. Near a ridge, k can be described as

$$\begin{aligned} & k(l_m, t_m, \theta) & (1) \\ = \; & (I \otimes M)(l_m, t_m) & (2) \\ = \; & c \exp(-\frac{t_m^2}{2\sigma_x^2})(b_0 + b_1 t_m + b_2 t_m l_m + b_3 t_m^2) & (3) \end{aligned}$$

with

$$\begin{aligned} c &= \frac{\sigma_b}{\sigma_x^3} \\ b_0 &= s_1 \hat{\sigma}_u^2 \sin\theta \\ b_1 &= h \cos\theta \\ b_2 &= s_1 \cos\theta \\ b_3 &= s_1 \frac{\hat{\sigma}_v^2 - \hat{\sigma}_u^2}{\sigma_x^2} \sin\theta \cos^2\theta \\ \hat{\sigma}_u^2 &= \sigma_u^2 + \sigma_b^2 \\ \hat{\sigma}_v^2 &= \sigma_v^2 + \sigma_b^2 \\ \sigma_x^2 &= \hat{\sigma}_u^2 \sin^2\theta + \hat{\sigma}_v^2 \cos^2\theta \end{aligned}$$

where l_m, t_m and θ are the position and the orientation of the mask with respect to the ridge.

So far, three coordinate systems have been introduced: the image coordinate system, with axes (x, y); the ridge local coordinate system, with axes (l, t); and the mask coordinate system, with axes (u, v). Besides these, there is one more coordinate system which will show up shortly, that is the relative coordinate system of the ridge, with axes $(\hat{x}, \hat{y})$. These four systems are essential not only for clarification, but also for reducing the computational complexity.

3.2 Estimating the Parameters

Obviously, Equation (3) is too complicated to deal with, and needs to be simplified. There are two directions of simplification. First, keep θ as small as possible, such that cosθ can be approximated as 1 and sinθ can be approximated as θ. Second, keep $|l_m|$ and $|t_m|$ less than 1. As mentioned in Section 2.1, the absolute ratio of h and s_1 is much larger than 20, so that when $|l_m|$ and $|t_m|$ are less than 1, the higher-order terms, $(s_1 \cos\theta)\, t_m l_m$ and $(s_1 \frac{\hat{\sigma}_v^2 - \hat{\sigma}_u^2}{\sigma_x^2} \sin\theta \cos^2\theta)\, t_m^2$, in (3) are negligible compared to $(h \cos\theta)\, t_m$.

Since θ is the orientation of the mask with respect to the ridge, in order to keep θ small, the orientation angle of the mask must be chosen close to the angle of the ridge. In the algorithm, there are four possible orientation angles for masks, which are 0, $\pi/4$, $\pi/2$, and $3\pi/4$. The two masks with orientation angles closest to the angle of the ridge will be used, which makes $|\theta| \leq \pi/4$ and makes the first simplification reasonable.

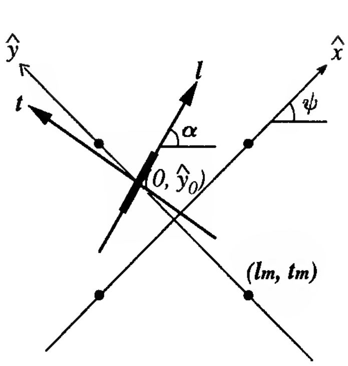


Figure 2: The transformation from the ridge local coordinates (l, t) to the relative coordinates $(\hat{x}, \hat{y})$.

To keep l_m and t_m small, the relative coordinates $(\hat{x}, \hat{y})$ mentioned in Section 3.1 need to be used here. The origin of relative coordinates is placed in the center of the grid square where the ridge is passing, then the axes of relative coordinates are rotated. There are also four possible rotation angles, 0, $\pi/4$, $\pi/2$, and $3\pi/4$. The one closest to the angle of the ridge will be used. Because the masks will be placed on the grids of image coordinates, choosing the masks whose positions are right on the vertices of this square will keep $|l_m|$ and $|t_m|$ less than $\sqrt{2}/2$. Therefore, it makes sense to eliminate the higher-order terms in (3), and to approximate $\exp(-\frac{t_m^2}{2\sigma_x^2})$ as 1.

By approximating cosθ, sinθ and $\exp(-\frac{t_m^2}{2\sigma_x^2})$, and then eliminating the terms with order higher than 2 in (3), the moment k can be simplified as:

$$\begin{aligned} k(l_m, t_m, \theta) &\approx c\,(b_0 + b_1 t_m) & (4) \\ &= c\,(s_1 \hat{\sigma}_u^2 \theta + h t_m) & (5) \\ &\approx c\; h\; t_m & (6) \end{aligned}$$

here $s_1 \hat{\sigma}_u^2 \theta$ is ignored from (5) to (6) because it is much less than $h t_m$. Typically, $|s_1|$ is much smaller than 0.1, $\hat{\sigma}_u^2$ is about 2, $|\theta|$ is smaller than $\pi/4$, $|h|$ is much bigger than 2, and $|t_m|$ is smaller than $\sqrt{2}/2$.

The parameters in (6) are evaluated with respect to the ridge local coordinates (l, t). They can be transferred into the relative coordinates $(\hat{x}, \hat{y})$ (see Fig. 2) by using the following transformation.

$$\begin{aligned} l_m &= \cos(\alpha - \psi)\, \hat{x} + \sin(\alpha - \psi)(\hat{y} - \hat{y}_0) & (7) \\ t_m &= -\sin(\alpha - \psi)\, \hat{x} + \cos(\alpha - \psi)(\hat{y} - \hat{y}_0) & (8) \end{aligned}$$

where α is the angle of the ridge, ψ is the rotation angle of the relative coordinates, and $(0, \hat{y}_0)$ is the intersection of the ridge and the y axis of relative coordinates.

Substituting (8) into (6), moment k can be expressed

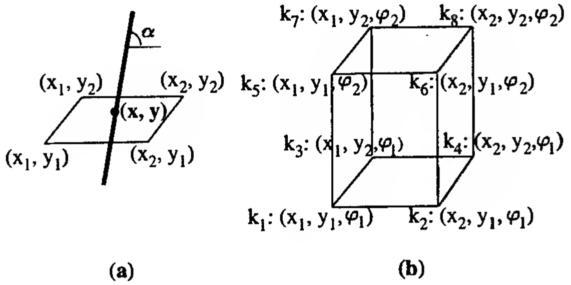

α
(x_1, y_2)
(x_2, y_2)
(x, y)
(x_1, y_1)
(x_2, y_1)
$k_7: (x_1, y_2, \varphi_2)$
$k_8: (x_2, y_2, \varphi_2)$
$k_5: (x_1, y_1, \varphi_2)$
$k_6: (x_2, y_1, \varphi_2)$
$k_3: (x_1, y_2, \varphi_1)$
$k_4: (x_2, y_2, \varphi_1)$
$k_1: (x_1, y_1, \varphi_1)$
$k_2: (x_2, y_1, \varphi_1)$
(a)
(b)

Figure 3: (a) A ridge; (b) The eight nearby points for the ridge.

as:

$$
\begin{aligned}
& k(\hat{x}, \hat{y}, \hat{\varphi}) && (9)\\
\approx\ & ch\,[-\sin(\alpha-\psi)\hat{x} + \cos(\alpha-\psi)(\hat{y}-\hat{y}_0)] && (10)\\
\approx\ & ch\,[-(\alpha-\psi)\hat{x} + (\hat{y}-\hat{y}_0))] && (11)
\end{aligned}
$$

where $\hat{x}$. $\hat{y}$ and $\hat{\varphi}$ are the position and the orientation of the mask with respect to the relative coordinates. If ψ is chosen to be closest to α among the four rotation angles, the maximum value of $|\alpha-\psi|$ will be $\pi/8$, such that $\sin(\alpha-\psi)$ can be approximated as $(\alpha-\psi)$, and $\cos(\alpha-\psi)$ can be approximated as 1, as shown from (10) to (11).

Let h_0 be equal to ch, and α_0 be equal to $(\alpha-\psi)$. In (11), the three unknown parameters, h_0, α_0, and $\hat{y}_0$, contain information about the amplitude, orientation, and position of a ridge respectively. In order to estimate these parameters, it is necessary to have at least three equations. Based on the criterion used in simplifying (3), given a ridge with angle α and location (x, y), the eight nearby moments should be adopted. Those eight points are shown in Fig. 3, with

$$
\begin{aligned}
x_1 &= \lfloor x \rfloor\\
x_2 &= \lceil x \rceil\\
y_1 &= \lfloor y \rfloor\\
y_2 &= \lceil y \rceil\\
\varphi_1 &= \lfloor \frac{\alpha}{\pi/4} \rfloor \times \frac{\pi}{4}\\
\varphi_2 &= \lceil \frac{\alpha}{\pi/4} \rceil \times \frac{\pi}{4}
\end{aligned}
$$

hence, there are eight equations with three unknowns:

$$k_i = h_0\ [-\alpha_0\ \hat{x}_i + (\hat{y}_i - \hat{y}_0)] \qquad i = 1 \ldots 8$$

To estimate the three unknown parameters, least-squares fitting is used. Let

$$\varepsilon = \sum_{i=1}^{8} \{k_i - h_0\,[-\alpha_0\ \hat{x}_i + (\hat{y}_i - \hat{y}_0)]\}^2 \qquad (12)$$

The set of $(\tilde{y}_0, \tilde{\alpha}_0, \tilde{h}_0)$ which minimizes ε will be found (see Section 4); that is, an edgel candidate will be detected.

3.3 Choosing the Angles

As shown in the previous section, there are two kinds of angles that need to be chosen in the algorithm. They are the rotation angle ψ of the relative coordinates, and the orientation angle φ_i of the mask. Note that

$$\psi \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$$

$$\varphi_1, \varphi_2 \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$$

Depending on the angle α of the targeted ridge, ψ is chosen to be the closest angle to α, and φ_1 and φ_2 are chosen to be the closest two angles to α. Since α is unknown before the detection, ψ, φ_1, and φ_2 are chosen on a trial-and-error basis by dividing the image plane into eight regions (see Table 1).

Table 1: the angle of the ridge (α), the rotation angle of the relative coordinates (ψ), and the orientation angles of the masks (φ_1, φ_2).

Region	α	ψ	φ_1, φ_2
1	$[0, \pi/8)$	0	$0, \pi/4$
2	$[\pi/8, \pi/4)$	$\pi/4$	$0, \pi/4$
3	$[\pi/4, 3\pi/8)$	$\pi/4$	$\pi/4, \pi/2$
4	$[3\pi/8, \pi/2)$	$\pi/2$	$\pi/4, \pi/2$
5	$[\pi/2, 5\pi/8)$	$\pi/2$	$\pi/2, 3\pi/4$
6	$[5\pi/8, 3\pi/4)$	$3\pi/4$	$\pi/2, 3\pi/4$
7	$[3\pi/4, 7\pi/8)$	$3\pi/4$	$3\pi/4, 0$
8	$[7\pi/8, \pi)$	0	$3\pi/4, 0$

In the algorithm, the ridge is first assumed to belong to region 1. In this case, ψ will be chosen as 0; φ_1 and φ_2 will be chosen as 0 and $\pi/4$ respectively. The operator discussed in the previous section is then used to determine the three unknown parameters of the ridge. Recall that one of the three unknowns is the orientation. If the ridge with that estimated orientation is in region 1, this means the hypothesis is correct, and the ridge is detected successfully. Otherwise, the result should be discarded, and the next region will be tested. For region j, $j = 1 \ldots 8$, the values of ψ, φ_1, and φ_2 in Table 1 are used.

4 Statistical Analysis

Because noise in an image causes spurious edgels and false alarms, it is useful to understand how noise affects the detections of edgels. The fluctuations of the edgel parameters provide not only information about how robust the algorithm is under noise, but also a mathematical basis for the linking.

As mentioned in Section 3.3, there are eight regions in total. Depending on which region the edgel belongs to, the least-squares solutions will be slightly different. But since the idea applied to each region is the same, the analyses for them are also similar. In this section, an edgel in region 1 will be discussed thoroughly. For edgels in the other regions, the results of theoretical analyses will be given in Appendix B.

From (12), the least-squares solutions for an edgel in region 1 are:

$$\begin{aligned}
\tilde{h}_0 &= -\frac{k_1+k_2+k_5+k_6-k_3-k_4-k_7-k_8}{4} \\
\tilde{\alpha}_0 &= -\frac{k_1+k_3+k_5+k_7-k_2-k_4-k_6-k_8}{k_1+k_2+k_5+k_6-k_3-k_4-k_7-k_8} \\
\tilde{y}_0 &= \frac{1}{2}\frac{k_1+k_2+k_3+k_4+k_5+k_6+k_7+k_8}{k_1+k_2+k_5+k_6-k_3-k_4-k_7-k_8}
\end{aligned}$$

where k_i is the convolution of the noisy image with the mask M_i.

Let $\Delta\alpha_0$ be the fluctuation of the orientation,

$$\begin{aligned}
\Delta\alpha_0 &= \tilde{\alpha}_0 - \alpha_0 \\
&= \sum_{i=1}^{8} \frac{\partial\alpha_0}{\partial k_i}\delta k_i + \text{H.O.T.} \\
&= \delta\alpha_0 + \text{H.O.T.}
\end{aligned}$$

where, $\delta\alpha_0$ is the linear approximation of $\Delta\alpha_0$.

Then, the standard deviation of $\tilde{\alpha}_0$ can be expressed as

$$\begin{aligned}
\sigma^2_{\tilde{\alpha}_0} &\approx Var(\delta\alpha_0) &\quad (13)\\
&= E\left[(\sum_{i=1}^{8}\frac{\partial\alpha_0}{\partial k_i}\delta k_i)^2\right] &\quad (14)\\
&= \sum_{i=1}^{8}\sum_{j=1}^{8}\left[\frac{\partial\alpha_0}{\partial k_i}\,\frac{\partial\alpha_0}{\partial k_j}E(\delta k_i \times \delta k_j)\right] &\quad (15)
\end{aligned}$$

In order to compute the variance of $\delta\alpha_0$, it is necessary to first calculate the variance of δk_i, and the covariance of δk_i and δk_j.

Recall that the moment k is the convolution of the noisy image with the mask M, and that the noise is modeled as the additive, zero-mean, white Gaussian function with standard deviation σ_N. Since the convolution is a linear operation, it can be deduced that the fluctuation of the moment, δk, will be a Gaussian random variable with mean and variance expressed as

$$\begin{aligned}
E(\delta k_i) &= 0 \\
Var(\delta k_i) &= E(\delta k_i \times \delta k_i) \\
&= \sigma_N^2 \int M_i(u,v)^2 du dv \\
&= \frac{1}{8\pi\sigma_u\sigma_v^3}\,\sigma_N^2
\end{aligned}$$

Similarly, the covariance of δk_i and δk_j can be derived from

$$E(\delta k_i \times \delta k_j) = \sigma_N^2 \int M_i(u,v) M_j(u,v) du dv$$

here $i = 1 \ldots 8$ and $j = 1 \ldots 8$. All of them will be listed in Appendix A.

Let σ_u and σ_v equal 1, and substitute all the variances and covariances into (15), the approximate fluctuation of the orientation, $\delta\alpha_0$, will be a Gaussian random variable with mean and variance expressed as

$$\begin{aligned}
E(\delta\alpha_0) &= 0 &\quad (16)\\
Var(\delta\alpha_0) &\approx (0.12 + 0.10\alpha_0 + 0.42\alpha_0^2)\frac{\sigma_N^2}{h^2} &\quad (17)
\end{aligned}$$

Using the same method, the approximate fluctuation of the position, $\delta\hat{y}_0$, will be a Gaussian random variable with mean and variance expressed as

$$\begin{aligned}
E(\delta\hat{y}_0) &= 0 &\quad (18)\\
Var(\delta\hat{y}_0) &\approx (0.2418 + 0.4249\hat{y}_0^2)\frac{\sigma_N^2}{h^2} &\quad (19)
\end{aligned}$$

Conventionally, the ratio of h and σ_N is called the signal to noise ratio, or the normalized contrast. From (17) and (19), it can be seen that edges with larger contrast have smaller fluctuations, and edges with smaller contrast have larger fluctuations.

These models of fluctuations have been verified by experiments. In the first step of simulation, synthesized images with edges in known positions and orientations were generated. Then the edge detector was applied to them to determine the edgel parameters. Finally, the standard deviations of $\tilde{\alpha}_0$ and $\tilde{y}_0$ were calculated.

All the simulation data and theoretical result are plotted in Fig. 4 and Fig. 5: the relation between $\sigma_{\tilde{\alpha}_0}$ (in radians) and contrast is plotted in Fig. 4; the relation between $\sigma_{\tilde{y}_0}$ (in pixels) and contrast is plotted in Fig. 5. In these figures, circles represent the sample variances of the simulation data, and line represents the expected values of $\sigma^2_{\tilde{\alpha}_0}$ and $\sigma^2_{\tilde{y}_0}$ over α_0 and $\hat{y}_0$ respectively, i.e.

$$\begin{aligned}
E_{\alpha_0}(\sigma^2_{\tilde{\alpha}_0}) &\approx E_{\alpha_0}[Var(\delta\alpha_0)] \\
&= 0.1449\,\frac{\sigma_N^2}{h^2} \\
E_{\hat{y}_0}(\sigma^2_{\tilde{y}_0}) &\approx E_{\hat{y}_0}[Var(\delta\hat{y}_0)] \\
&= 0.2772\,\frac{\sigma_N^2}{h^2}
\end{aligned}$$

here, α_0 is assumed to be uniformly distributed between 0 and $\pi/8$, and $\hat{y}_0$ is assumed to be uniformly distributed between -0.5 and 0.5.

It is obvious that the simulation data do match with the theoretical result. Furthermore, this algorithm is robust. It can detect the edges accurately even under large noise. For example, when the contrast equals 4, the standard deviation of $\tilde{\alpha}_0$ is about 2^{-3} radians, and the standard deviation of $\tilde{y}_0$ is about $2^{-2.8}$ pixels.

The simulation data (cross mark) and theoretical result (dash line) of the Wang-Binford operator are also

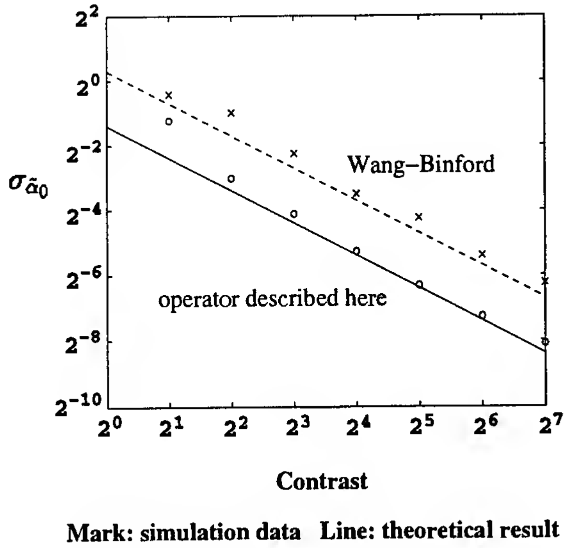


Figure 4: $\sigma_{\tilde{\alpha}_0}$ (in radians) versus contrast. The performance of the Wang-Binford operator is included as a comparison.

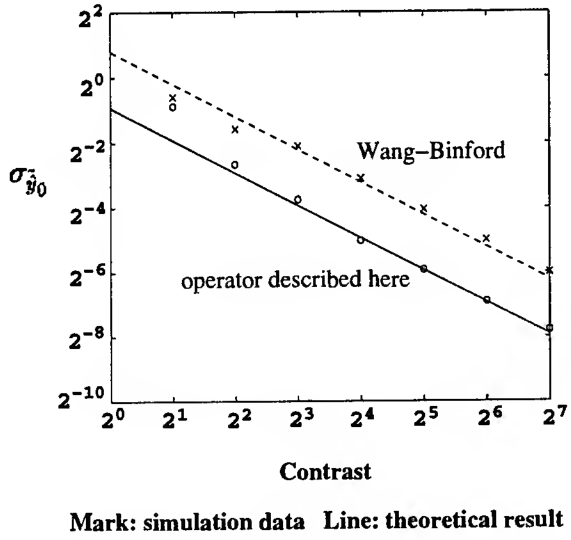


Figure 5: $\sigma_{\tilde{y}_0}$ (in pixels) versus contrast. The performance of the Wang-Binford operator is included as a comparison.

plotted in the figures as baselines for comparison. The standard derivations of both orientation and position have been improved by a factor of four in the edge detector described here.

5 Edge Images

5.1 Linking

The detected edgels in an image contain little information about the scene unless they are linked into extended edges. Previous work did little about linking due to the lack of accurate edgel data [8] [9]. In this algorithm, since the estimates of edgels are well done, the linking process becomes much easier.

Assume there are two nearby edgels with parameters (x_1, y_1, α_1) and (x_2, y_2, α_2), respectively. The hypothesis used here is that two edgels belong to the same extended edge. If it is true, the intensities of two edgels should have the same sign and the difference of their orientations should be small. Statistically, the hypothesis may be rejected when it is true. The probability of it is called the false negative rate. The false negative rate depends on the threshold [10]. As discussed in the previous section, the fluctuation of the edgel orientation is approximated by a Gaussian random variable, thus the threshold of α can be properly chosen to allow a constant false negative rate. In order to have the rate equal to 0.01, the threshold is set to be 2.8 σ_α. i.e. the acceptance region of the hypothesis test is

$$|\alpha_1 - \alpha_2| < 2.8\, \sigma_\alpha$$

From Section 4 and Appendix B, the fluctuations of the edgel orientation in different regions are derived. By averaging all of them, the variance of α can be written as

$$E(\sigma_\alpha^2) \approx 0.2554\, \frac{\sigma_N^2}{h^2}$$

Combining these two equations, the criterion used to group two edgels is

$$\begin{aligned} |\alpha_1 - \alpha_2| &< 2.8\, \sigma_\alpha \\ &= 2.8 \times (0.5053\, \frac{\sigma_N}{h}) \\ &\approx \frac{\pi}{2} \frac{\sigma_N}{h} \end{aligned}$$

That is, two nearby edgels will be linked together if the difference of their orientations is smaller than $\frac{\pi}{2}\frac{\sigma_N}{h}$.

The searching region is defined in Fig. 6. The pixel in which the current detected edgel sits is marked by a dot. The other 12 pixels in the searching region are marked by numbers. The numbers indicate the testing orders which are based on the distances between the testing pixels and the current pixel. The shorter the distance is, the higher priority the testing pixel has. In other words, the pixel with a lower number will be tested first. Note that it is not necessary to test the pixels righter or lower than the currently working

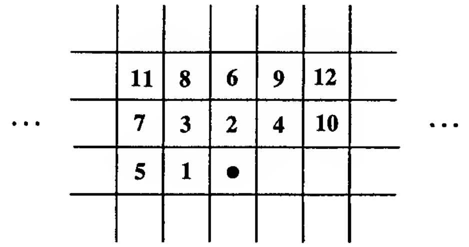


Figure 6: The searching region for linking. The dot marks the pixel in which the current detected pixel sits. The numbers mark the pixels which will be tested.

pixel because the tests will be done when the operator moves to those pixels.

To sum up, the operator scans the whole image once, from left to right, then from top to bottom. If it detects an edgel, it will test the pixels in the searching region. As soon as the edgel in the testing pixel meets the linking criterion, it will be grouped with the detected edgel, and the tests will stop. If all 12 tests fails, the detected edgel will become the beginning of a new edge.

5.2 Results

The detected edge images are superior even though the linking criterion used here is straightforward. Examples are given in Fig. 7, Fig. 8, Fig. 9, and Fig. 10. Fig. 7 shows the detected delta edges image of a natural scene; Fig. 8 shows the detected step edge image of roads Fig. 9 shows the detected step edge image of buildings; and Fig. 10 shows the detected crease edge image of a thread. The results indicate that the operator detects all three types of edges successfully and extracts the orientations and positions of edges correctly. Moreover, it takes only 20 seconds to detect a 256x256 image on an SGI Indy R4400SC workstation.

6 Conclusion

An edge detector for delta edges, step edges, and crease edges has been developed. It is insensitive to shading and noise, and it is fast.

Based on the physics of image formation, an edgel is modeled by three parameters, the position, orientation and amplitude. To extract information from the input image, the image is first convolved with a mask. By sampling the convolved image at each pixel at four orientations, the 2x2x2 cubes are built. The three edgels parameters are estimated over the cube; that is, the least-squares estimate of an edgel is obtained from the eight simplified equations under the relative coordinate system.

Theoretical analyses have been verified by simulation. It is shown that the algorithms can detect edges well even when the signal to noise ratio is less than 4.

Though only delta edges have been discussed thoroughly in this paper, the other two kinds of edges can be detected successfully by the same family of operators, as shown in Section 5.2. The results indicates that the extended edges are very accurate and informative.

The speed of the detector is fast. It takes 20 seconds to detect a 256x256 image on an SGI Indy R4400SC workstation.

Appendix A

The covariances for the adjacent eight moments k_i, $i = 1 \ldots 8$ (see Fig. 3(b)).

Let $\sigma_u = \sigma_v = \sigma$,

$$\begin{aligned} & E(\delta k_1 \times \delta k_2) = E(\delta k_3 \times \delta k_4) \\ = \; & E(\delta k_5 \times \delta k_6) = E(\delta k_7 \times \delta k_8) \\ = \; & \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{4\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_3) = E(\delta k_2 \times \delta k_4) \\ = \; & E(\delta k_5 \times \delta k_7) = E(\delta k_6 \times \delta k_8) \\ = \; & \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{4\sigma^2})(1 - \frac{1}{2\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_4) = E(\delta k_2 \times \delta k_3) \\ = \; & E(\delta k_5 \times \delta k_8) = E(\delta k_6 \times \delta k_7) \\ = \; & \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{2\sigma^2})(1 - \frac{1}{2\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_5) = E(\delta k_2 \times \delta k_6) \\ = \; & E(\delta k_3 \times \delta k_7) = E(\delta k_4 \times \delta k_8) \\ = \; & \frac{1}{\sqrt{2}} \sigma_N^2 \frac{1}{8\pi\sigma^4} \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_6) = E(\delta k_2 \times \delta k_5) \\ = \; & E(\delta k_3 \times \delta k_8) = E(\delta k_4 \times \delta k_7) \\ = \; & \frac{1}{\sqrt{2}} \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{4\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_7) = E(\delta k_2 \times \delta k_8) \\ = \; & E(\delta k_3 \times \delta k_5) = E(\delta k_4 \times \delta k_6) \\ = \; & \frac{1}{\sqrt{2}} \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{4\sigma^2})(1 - \frac{1}{2\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_8) = E(\delta k_4 \times \delta k_5) \\ = \; & \frac{1}{\sqrt{2}} \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{2\sigma^2})(1 - \frac{1}{\sigma^2}) \end{aligned}$$

$$\begin{aligned} & E(\delta k_1 \times \delta k_6) = E(\delta k_2 \times \delta k_7) \\ = \; & \frac{1}{\sqrt{2}} \sigma_N^2 \frac{1}{8\pi\sigma^4} \exp(-\frac{1}{2\sigma^2}) \end{aligned}$$

Appendix B

The expected values of the theoretical analyses for edgels in region i, $i = 1 \ldots 8$ (see Table 1).

Let $\sigma_u = \sigma_v = 1$,

$$\begin{array}{rrcl}
\text{region1}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.1449\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2772\,\frac{\sigma_N^2}{h^2} \\
\text{region2}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.1689\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2511\,\frac{\sigma_N^2}{h^2} \\
\text{region3}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.3151\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2511\,\frac{\sigma_N^2}{h^2} \\
\text{region4}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.4457\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2517\,\frac{\sigma_N^2}{h^2} \\
\text{region5}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.4072\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2517\,\frac{\sigma_N^2}{h^2} \\
\text{region6}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.2874\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2552\,\frac{\sigma_N^2}{h^2} \\
\text{region7}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.1678\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2552\,\frac{\sigma_N^2}{h^2} \\
\text{region8}: & E(\sigma^2_{\tilde{\alpha}_0}) & \approx & 0.1064\,\frac{\sigma_N^2}{h^2} \\
 & E(\sigma^2_{\tilde{y}_0}) & \approx & 0.2772\,\frac{\sigma_N^2}{h^2}
\end{array}$$

References

[1] A. Herskovits, and T. O. Binford, "On Boundary Detection," *M.I.T. Artificial Intelligence Lab., AI Memo 183*, Cambridge, Mass., 1970.

[2] B. K. P. Horn, "The Binford-Horn Line-Finder," *M.I.T. Artificial Intelligence Lab., AI Memo 285*, Cambridge, Mass., 1971.

[3] D. C. Marr, and E. C. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London B*, Vol. 204, 1980.

[4] J. F. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.

[5] V. S. Nalwa, and T. O. Binford, "On Detecting Edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 699-714, 1986.

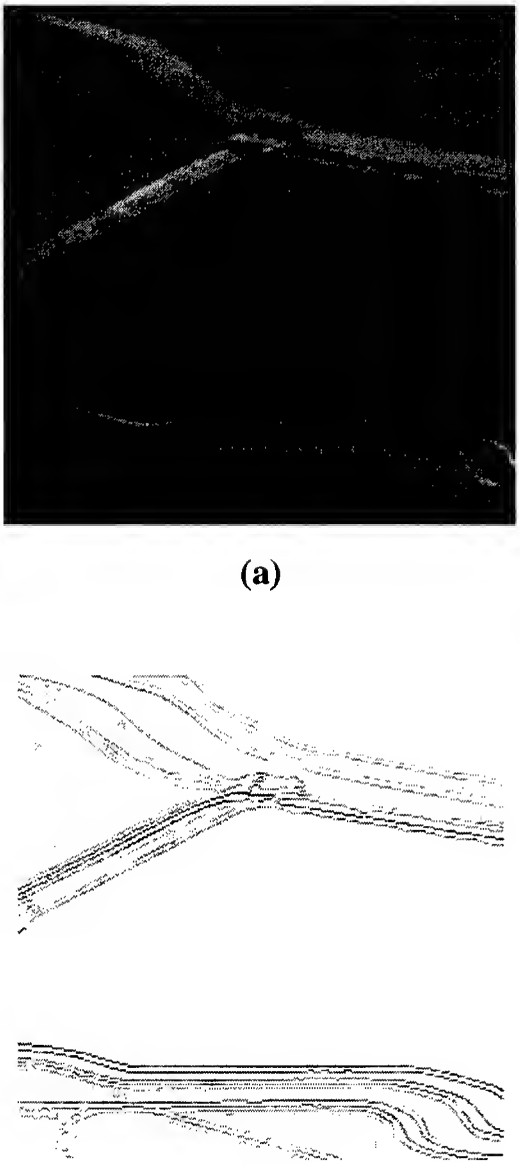
[6] Sheng-Jyh Wang, and T. O. Binford, "Generic, model-based estimation and detection of discontinuities in image surfaces," *Proceedings of 23rd Image Understanding Workshop*, Monterey, CA., Vol. 2, pp. 1443-1449, 1994.

[7] Sheng-Jyh Wang, and T. O. Binford, "Model-Based Edgel Aggregation," *Proceedings of 23rd Image Understanding Workshop*, Monterey, CA., Vol. 2, pp. 1589-1593, 1994.

[8] R. Nevatia, and K. R. Babu, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing*, Vol. 13, pp. 257-269, 1980.

[9] V. S. Nalwa, and E. Pauchon, "Edge Aggregation and Edge Description," *Computer Vision, Graphics, and Image Processing*, Vol. 40, pp. 79-94, 1987.

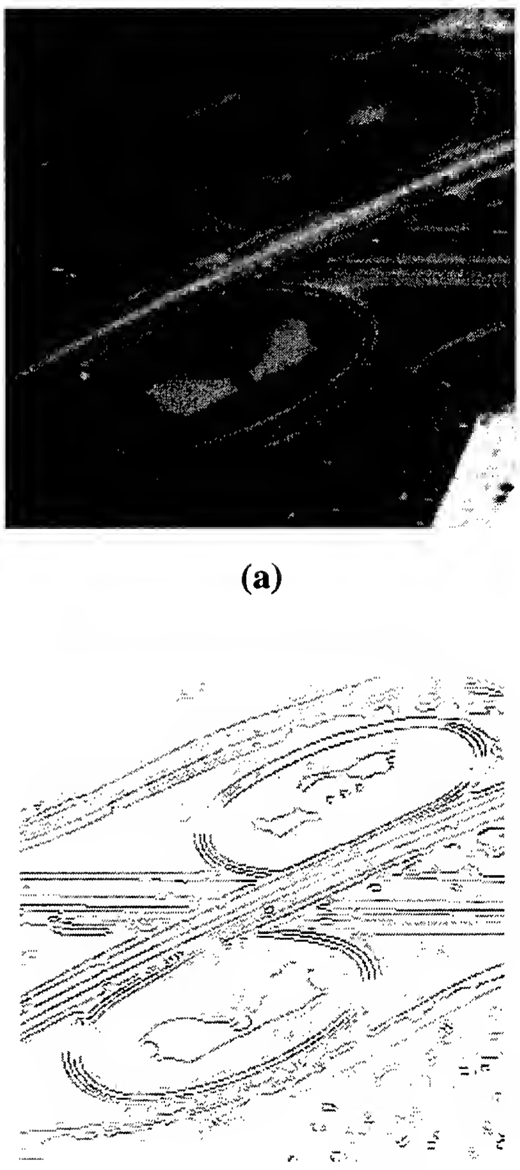
[10] John A. Rice, "Mathematical Statistics and Data Analysis," Duxbury Press, pp. 300-306, 1995.



(a)

(b)

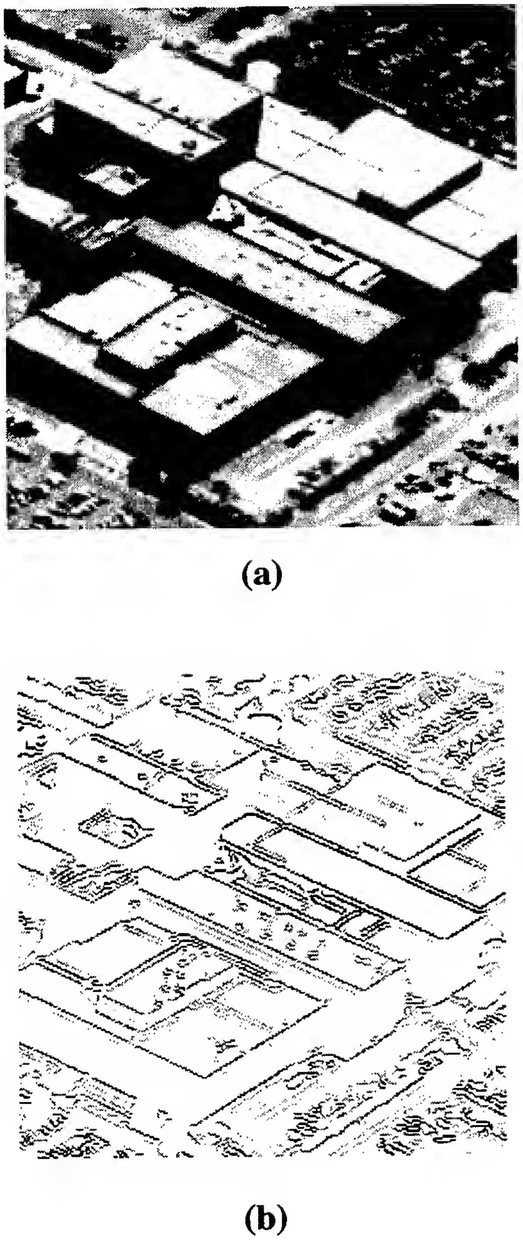
Figure 7: (a) The original image (256 x 256); (b) The detected delta edge image.



(a)

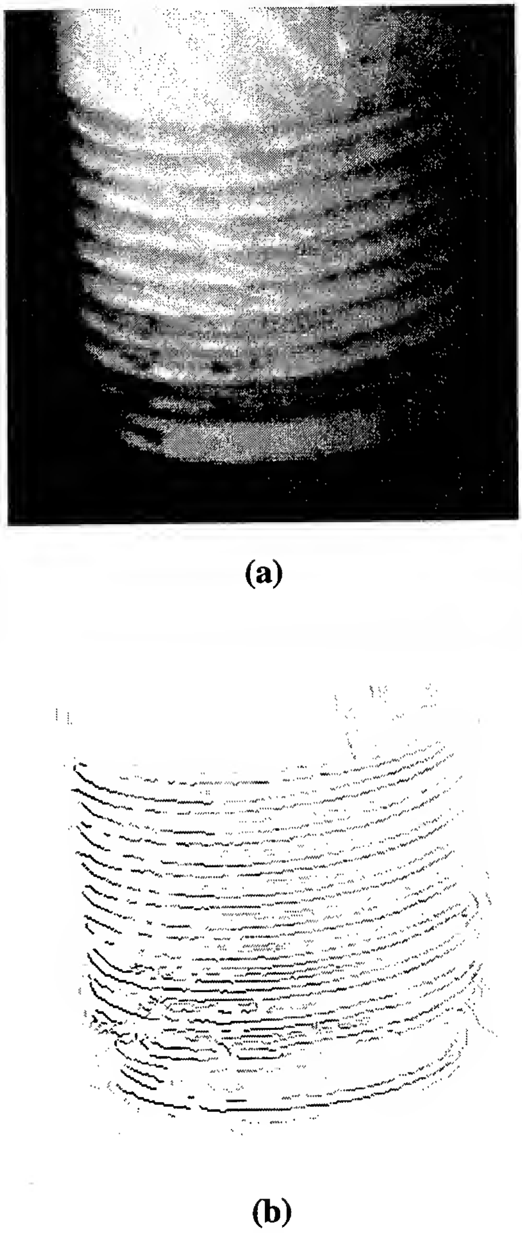
(b)

Figure 8: (a)The original image (256 x 256);(b)The detected step edge image.


(a)

(b)

Figure 9: (a) The original image (256 x 256); (b) The detected step edge image.


(a)

(b)

Figure 10: (a) The original image (135 x 136); (b) The detected crease edge image.

Automated Construction of Templates for Matching

Gang Liu **Robert M. Haralick**
Intelligent Systems Laboratory
Department of Electrical Engineering
University of Washington
Box 352500
Seattle, WA 98195-2500
E-MAIL: {gliu,haralick}@george.ee.washington.edu

Abstract

This paper presents a methodology for automatically extracting a set of positions from the image of a target to assemble a template, which can enable high accuracy detection and localization of the target in imagery using template matching based object recognition algorithms.

The selection criteria is based on determining the covariance of the location produced by the template matching algorithm. The methodology minimizes this covariance by selecting the right set of positions. Using only subsets of positions in the object can increase the invariance of the template, and has the potential of improving the robustness of the template matching algorithm to non-additive, non-Gaussian noise. This methodology can also be generalized to apply to many distance based recognition algorithms.

1 Introduction

The difficulty with the classical use of templates for target recognition is two fold: one has to do with the spatial perturbation of the target in the image and the other has to do with the fact that the gray scale perturbing noise is not additive and non-Gaussian. Therefore when the template matching is done or when the matched filtering is done, the quantity computed is not related to the log probability of the data given the target. In this paper, we try to deal with the second effect, the effect of non-additive non-Gaussian perturbation.

When designing a particular template matching algorithm, there are two things to be done first: obtaining the template and devising the similarity measure. Conventional template matching algorithm development is like an "open-loop" system, where identifying the template is always the first step and after that the obtained template is held optimal and no update is intended for it. Our methodology "closes the loop" in the algorithm development stage by examining the location produced by a choice of the template (and its corresponding similarity measure), and using the obtained information to modify the template so that the new template will produce a better estimate of the location. This is achieved by analytically propagating the perturbation on the image data through to the covariance of the location estimated for the position of the target. In the particular context of target detection and recognition, we start with some small initial template and find the spatial direction in which the estimated location has the highest variance. Then we examine other parts of the target area and add into the template those areas such that the resulting template will give a location estimate whose variance in that direction is the smallest.

Our discussion addresses the simplest example where the templates are sets of pixel positions with their gray level values, and the similarity measure (actually a difference measure) is the mean squared error. We also choose the 1-D model to simplify the notation. However, the templates can be composed of other more general feature values such as the coefficients of some orthogonal transforms of the image of the object. Therefore, this methodology is not restricted to the example presented here. Since many of the existing matching based algorithms follow this formulation and their similarity measures are usually fairly standard, the methodology we present can be applied to a variety of algorithms to improve their performance.

2 Subset versus entire object

Template matching using plain pixel values is mostly used in the cases where the interested objects do not have many outstanding features, or where such features cannot be detected reliably. Conventional template matching methods use all positions and their pixel values inside the object of interest. The idea behind this practice is to use as much information as possible about the object. Most often, pixels in the template are treated equal in their contribution in the similarity measure.

However, using target areas which are spatially non-distinct and have low contrast not only do not contribute, but in effect can negatively affect detection accuracy and localization. The areas of the target that are least affected by non-additive non-Gaussian noise are those target areas in which there is a relatively high contrast spatial structure. The high contrast will be the dominant effect relative to the non-additive non-Gaussian noise perturbation and thereby permit good detection even with a technique whose basis assumes an additive Gaussian perturbation. Assembling these spatial structures together as the template will permit good localization of the position of the detected target.

In the next section, we introduce an idea of measuring the *discriminatory power* of a template. Although the discriminatory power of the template is the biggest when all points are included, the major share of the discriminatory power is often contributed by a small subset of the points, and the contribution from the rest of the points is marginal. Some of the points from which most of the discriminatory power comes will be chosen to form the so called most discriminatory set. When this set is used to represent the original entire object, the deviation of the assumed noise model from the true distribution of the noise will have the least effect on the process of locating the object. Thus, although a small amount of the discriminatory power is lost by not using all the points, this disadvantage is overcome by the increased robustness to noise for which the assumed model is wrong.

3 Finding the best subset

3.1 A measure of discriminatory power of templates

When template matching is used to detect and locate a certain object in a noisy observation, the result produced is usually perturbed from the true location. This perturbation is called the output perturbation, and is a function of both the template and the perturbation present in the observation which is called the input perturbation. This output perturbation constitutes the error of the output from the template matching algorithm, and is a random variable usually with mean zero. The covariance propagation technique [Haralick, 1996] can be used to express the covariance of the error as a function of the covariance of the input perturbation. This function is a good measure of how sensitive the location of this template is to the input perturbation. When the variance of the input perturbation is kept at a fixed level, this sensitivity is high if the variance of the error is large, and hence we say the discriminatory power of the template is low. If the variance is small, the sensitivity is low, so the template is relatively easier to detect and locate in the perturbed observation, therefore it has more discriminatory power.

Consider the example of a 1-D template in Figure 1. We want to use the MSE as the similarity measure to detect this pattern in some perturbed input. Intuitively, some points in the neighborhood of low contrast could be taken away from the template without significantly lowering the performance of matching. However, if one or two of the large values were taken

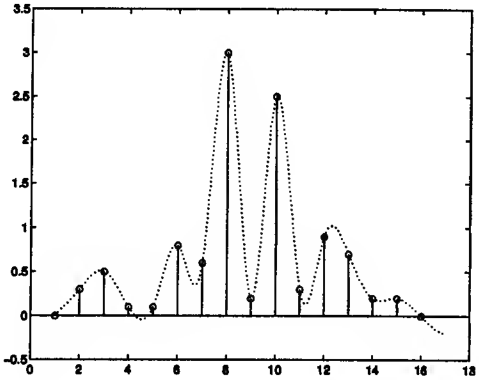


Figure 1: An example of a discrete 1-d template. Dotted line is the cubic spline interpolating function.

away, the uncertainty of the result should increase significantly.

We did an experiment to see if this intuition is correct. We generated a 100-point *iid* $N(0,1)$ noise. Then we multiplied the template in Figure 1 by 2.3 and added it to the noise, with its first point aligned with point #43 of the noise. This is one realization of the perturbed observation. Then we used exhaustive search to find the estimate of the template's location that minimizes the MSE. This procedure was repeated 100,000 times and the sample mean and variance of the difference between the estimate and the true location were obtained. The sample variance in this case was 1.28×10^{-3}. Then we did two more experiments with the point #3 and #10 taken out from the original template respectively. The sample variances were 7.09×10^{-3} and 8.15×10^{-1}, respectively. So taking out point #3 did not matter too much, but taking out point #10 caused big drop in the performance. This result confirmed our intuition that points in a template differ in their contribution to the discriminatory power of the template.

In the next sub-sections, we will use covariance propagation to get a theoretical prediction of the output variance as a function of the input variance.

3.2 Notation

Since we will be applying the technique to target recognition in digital images, we discuss the problem in the discrete case.

A template consists of a set of positions, each of which is associated with the signal value of the template at that position. Let $X = \{x_1, x_2, \ldots, x_N\}(x_1 < x_2 < \ldots < x_N)$ be a set of positions. The position of the n-th point of the template relative to the chosen reference point is x_n. The signal value of the point is $h(x_n)$. Let $H = (h(x_1), h(x_2), \ldots, h(x_N))^T$ be an $N \times 1$ vector denoting the template.

Suppose our observation is made on a finite set of M discrete positions. This set is denoted as $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_M\}$. For each position γ_m we observe a signal value $s(\gamma_m)$, where $s(\cdot)$ is a real function defined on Γ. Let $S = (s(\gamma_1), s(\gamma_2), \ldots, s(\gamma_M))^T$ be an $M \times 1$ vector for the observed values, which we call the observable.

Since X and Γ are both discrete point sets, there is only a finite set of locations where the template can possibly appear in the observable. (This restriction could be relaxed if we would estimate the location of the template to subpixel accuracy, but we choose not to do that and require X and Γ to be on the same sampling grid.) We use θ to denote the location of the reference point of the template in the observable, then the set of possible θ is

$$\Theta = \{\theta | X_\theta \subseteq \Gamma\} \tag{1}$$

where $X_\theta = \{x + \theta | x \in X\}$.

In usual digital signal applications, Γ is given by the sampling process as an evenly spaced location set. X is a chosen set whose spacing might be uneven, but the spacing between each successive pair is always some multiple of the spacing in Γ. When these two are given, Θ is uniquely determined by the above equation.

Suppose the template is present in the observable at location θ_0, (that is, with its reference point located at θ_0), where θ_0 is a fixed constant from Θ. In the case of ideal observation where there is no perturbation, we use $s^*(x)$ to denote the value of the observable at position

$x \in \Gamma$. The template is just a part of the ideal observable, and

$$h(x) = s^*(x + \theta_0) \qquad \text{for } x \in X \qquad (2)$$

With additive *iid* Gaussian perturbation, we will observe

$$\hat{s}(x) = s^*(x) + \nu(x) \qquad x \in \Gamma \qquad (3)$$

where the perturbation $\nu(\cdot)$ is defined on Γ, and $\nu = (\nu(\gamma_1), \nu(\gamma_2), \ldots, \nu(\gamma_M))^T \sim N(0, \sigma^2 I_M)$, where I_M is the $M \times M$ identity matrix, and σ^2, a positive scalar, is the variance of each element of ν.

We use $S^* = (s^*(\gamma_1), s^*(\gamma_2), \ldots, s^*(\gamma_M))^T$ and $\hat{S} = (\hat{s}(\gamma_1), \hat{s}(\gamma_2), \ldots, \hat{s}(\gamma_M))^T$ to denote the observations made in these two cases, and they are related by $\hat{S} = S^* + \nu$.

3.3 Criterion function

The MSE criterion function is used and to be minimized with respect to the location parameter θ.

$$F(S, \theta) = \sum_{n=1}^{N} [s(x_n + \theta) - h(x_n)]^2 \qquad (\theta \in \Theta) \quad (4)$$

In the ideal no perturbation case, $S = S^*$ and $F(S^*, \theta_0) = 0$. Therefore the true location θ_0 is the solution to the problem in the ideal case. When there is perturbation on the observable, we get $S = \hat{S}$ and the estimated location is

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \left\{ \sum_{n=1}^{N} [\hat{s}(x_n + \theta) - h(x_n)]^2 \right\} \qquad (5)$$

There are various schemes for the minimization. Our requirement is that the solution $\hat{\theta}$ from the chosen technique is a local minimum of $F(S, \theta)$, i.e.,

$$\frac{\partial F(S, \theta)}{\partial \theta} = 0 \qquad (6)$$

at $(\hat{S}, \hat{\theta})$. Equation 6 also holds for (S^*, θ_0) since it is a global minimum of Equation 4.

3.4 Propagating the covariance

Suppose that given observable $S = \hat{S}$, from some minimization scheme we have obtained a $\hat{\theta}$ that satisfies Equations 5 and 6. Due to the perturbation on $\hat{S}$, $\hat{\theta}$ is also perturbed from θ_0. Assuming this perturbation is small and also additive, we obtain $\hat{\theta} = \theta_0 + \Delta\theta$ where $\Delta\theta$ is the small perturbation.

Consider, for $\theta \in \Theta$,

$$\begin{aligned} g(S, \theta) &= \frac{\partial F(S, \theta)}{\partial \theta} \\ &= \sum_{n=1}^{N} 2[s(x_n + \theta) - h(x_n)] \cdot \frac{ds(x_n + \theta)}{d\theta} \qquad (7) \end{aligned}$$

In particular, we want to study the above function at two particular points: (S^*, θ_0) and $(\hat{S}, \hat{\theta})$, namely the ideal noise-free observable with the true location of the template, and the perturbed observable with the estimated location from our selected minimization scheme.

Assuming that the perturbations ν and $\Delta\theta$ are small, we approximate $g(\hat{S}, \hat{\theta})$ by the linear terms of its Taylor series expansion at (S^*, θ_0). So, approximately, we write

$$\begin{aligned} g(\hat{S}, \hat{\theta}) \;&=\; g(S^*, \theta_0) + \left(\frac{\partial g(S^*, \theta_0)}{\partial S} \right)^T \cdot \nu \\ &\quad + \left(\frac{\partial g(S^*, \theta_0)}{\partial \theta} \right)^T \cdot \Delta\theta \qquad (8) \end{aligned}$$

where we have used the notation

$$\frac{\partial g(S^*, \theta_0)}{\partial S} = \left. \frac{\partial g(S, \theta)}{\partial S} \right|_{\substack{S = S^* \\ \theta = \theta_0}}$$

$$\frac{\partial g(S^*, \theta_0)}{\partial \theta} = \left. \frac{\partial g(S, \theta)}{\partial \theta} \right|_{\substack{S = S^* \\ \theta = \theta_0}}$$

Since both (S^*, θ_0) and $(\hat{S}, \hat{\theta})$ are local minima of $F(S, \theta)$ in Equation 4, we have

$$g(S^*, \theta_0) = 0 \quad \text{and} \quad g(\hat{S}, \hat{\theta}) = 0$$

Let

$$A = \frac{\partial g(S^*, \theta_0)}{\partial \theta} \quad \text{and} \quad B = \frac{\partial g(S^*, \theta_0)}{\partial S}$$

and assuming that A^{-1} exists, we obtain

$$\Delta\theta = -\left(A^{-1} \right)^T \cdot B^T \cdot \nu \qquad (9)$$

Recall that $\nu \sim N(0, \sigma^2 I_M)$, we have

$$\begin{aligned}\mathbf{E}(\Delta\theta) &= 0 \\ \mathbf{Cov}(\Delta\theta) &= \sigma^2 \cdot \left(A^{-1}\right)^T \cdot B^T \cdot B \cdot A^{-1}\end{aligned}$$

Since θ only takes value from the discrete set Θ, the derivative $ds(x_n + \theta)/d\theta$ in Equation 7 is undefined. We have to define it properly before we can get an expression of $\mathbf{Cov}(\Delta\theta)$ in terms of H. Here we use cubic spline interpolation [Press *et el.*, 1992] to obtain a continuous interpolating function $\tilde{s}(\cdot)$ from S. The $d\tilde{s}(x)/dx$ is used in place of $ds(x)/dx$. After some manipulations[Liu and Haralick, 1997], the final result is

$$\begin{aligned}&\mathbf{Cov}(\Delta\theta) = \sigma^2 \cdot \\ &\left\{\sum_{n=1}^{N-1} \left(\frac{h(x_{n+1}) - h(x_n)}{x_{n+1} - x_n} \right.\right. \\ &\quad \left. - \frac{x_{n+1} - x_n}{6}\left[h''(x_{n+1}) + 2h''(x_n)\right]\right)^2 \\ &\quad + \left(\frac{h(x_N) - h(x_{N-1})}{x_N - x_{N-1}} + \right. \\ &\quad \left.\left. \frac{x_N - x_{N-1}}{6}\left[2h''(x_N) + h''(x_{N-1})\right]\right)^2\right\}^{-1} \qquad (10)\end{aligned}$$

where $h''(x) = s^{*''}(x + \theta_0)$ for $x \in X$ and can be expressed in terms of S^*.

3.5 Choosing subset according to propagated covariance

Continuing the discussion in Section 3.1, Equation 10 can be used to measure the discriminatory power of the template H. When terms are taken out from the sum in Equation 10, the value of $\mathbf{Cov}(\Delta\theta)$ will increase, indicating less discriminatory power.

Suppose we have already chosen an initial template with a small number of points and want to add more points to it to increase the discriminatory power, but do not want to include points with marginal discriminatory power. One by one we examine the points not yet in the template by adding only that point to the template and calculate the output variance. The point yielding the smallest variance is chosen to be part of the template. This process is repeated until the output variance drops below a specified level, indicating enough discriminatory power has been achieved.

4 Experiment

Experiments using synthetic data were conducted to test the relationship between the variances of input and output perturbation obtained in the previous section.

4.1 Observable, template, and test statistics

We chose a cubic polynomial curve as the ideal observable, a segment of which is chosen as the template.

$$\begin{aligned}s^*(x) &= 0.05x^3 + 0.04x^2 - 4x + 2 \\ &\qquad (-10 \le x \le 10) \\ h(x) &= s^*(x) \qquad (-2 \le x \le 2)\end{aligned}$$

i.e., $\theta_0 = 0$. We sampled them on the evenly spaced sampling locations with sampling interval 0.1. The resulting observable has 201 points and the template has 41 points. By plugging the equation of the sampled template into Equation 10, we obtained the relationship between the output and input variances as $\mathbf{Cov}(\Delta\theta) \approx (1.69 \times 10^{-3}) \times \sigma^2$, where σ^2 is the variance of each element of the input noise vector ν.

For a fixed level of input noise, specified by the variance σ^2, a 201-dimensional *iid* Gaussian noise vector ν was generated and added to the ideal observable S^* to simulate the perturbed observable $\hat{S}$. Then an exhaustive search was performed to find the location $\hat{\theta}$ that minimizes the term in Equation 4. The error $\Delta\theta = \hat{\theta} - \theta_0$ was recorded.

Since the exhaustive search was done on a discrete set of locations, it had a finite spatial resolution. Let the sampling interval be q. In order for the quantization error to be small, the input noise level σ^2 should be in the range such that q is much less than σ_{out}, the standard deviation of the output distribution. It is desirable that q is in the range of $\frac{6}{100}\sigma_{out}$ to $\frac{6}{50}\sigma_{out}$. We held q fixed in the experiments to be 0.1. Using the re-

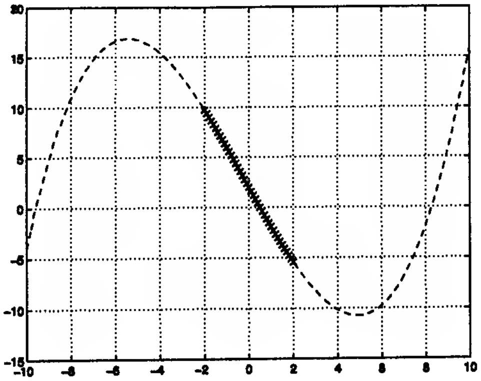


Figure 2: Ideal observable and template used in the experiment. Dashed curve is the ideal observable, and the 'x' marked part is the template. This continuous signal was evenly sampled with an interval of 0.1

lationship $\sigma_{out}^2 \approx 0.00169\sigma^2$, the desired range of q requires σ^2 to be in the range of approximately 400 to 1600. This noise perturbation is too large relative to our signal, and our assumption of small noise perturbation was very much not satisfied. So we have to sacrifice some spatial resolution in the experiment and take into account this quantization in interpreting and using the statistics gathered from the experiments. We take this into account by calculating the expected variance using Equation 10, setting this variance as the variance parameter of a continuous Gaussian distribution and then quantizing this distribution by our sampling interval. This produces a discrete probability distribution with an underlying continuous Gaussian distribution. The variance of this discrete probability distribution is used as the refined theoretical prediction of the output variance.

The noise generation and exhaustive search were repeated for 20,000 times and the sample distribution of $\Delta\theta$ was obtained. This sample distribution has mean 0 and variance $0.00169\sigma^2$.

With Gaussian random variables, the sufficient statistics are $\overline{\Delta\theta}$ and $S_{\Delta\theta}^2$, namely the sample mean and sample variance of $\Delta\theta$. Normalizing them with the predicted output variance, we obtain the statistics

$$T_1 = \frac{\overline{\Delta\theta}}{\sqrt{\mathbf{Cov}(\Delta\theta)/L}}$$

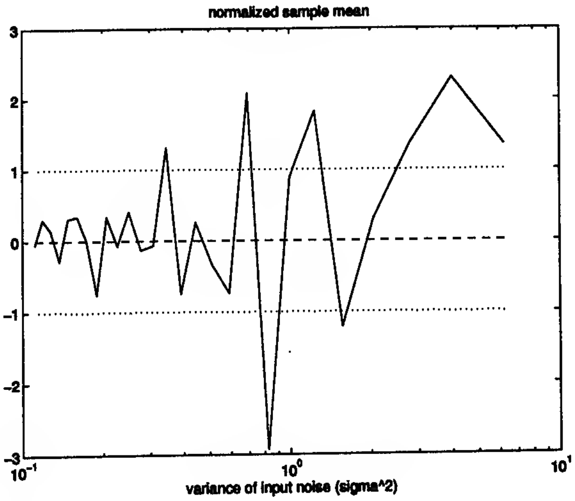


Figure 3: Plot of T_1, the normalized sample mean. $T_1 \sim N(0,1)$ with mean 0 (plotted in dashed line) and standard deviation 1 (plotted in dotted line around the mean).

$$T_2 = \frac{(L-1)S_{\Delta\theta}^2 + L\overline{\Delta\theta}^2}{\mathbf{Cov}(\Delta\theta)}$$

where $L = 20{,}000$ is the number of samples observed in the experiment. Under H_0, $T_1 \sim N(0,1)$ and $T_2 \sim \chi_{L-1}^2$. Their distributions are unknown when H_0 is not true.

A series of experiments was conducted for different levels of input noise where samples of $\Delta\theta$ were gathered and the statistics calculated. The normalized sample means are plotted in Figure 3. Figure 4 shows one example of the observed sample distribution, predicted discrete distribution, and the underlying predicted continuous Gaussian distribution for input noise variance $\sigma^2 = 2.78$. In this particular case, the sample values of $\Delta\theta$ are actually quantized into 8 bins. The sample mean is 6.55×10^{-4}, the sample variance is 7.42×10^{-3}, the predicted mean is 0, and the predicted variance is 4.70×10^{-3} for the underlying continuous Gaussian distribution and 5.50×10^{-3} for the quantized distribution.

4.2 Summary of the observed data

Figure 3 shows the agreement on the mean value of the output perturbation between the exper-

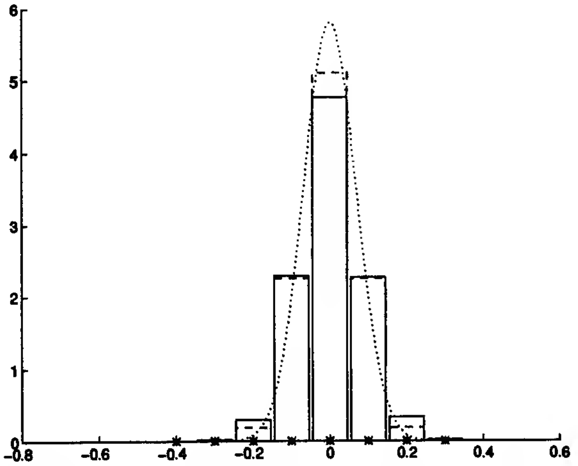


Figure 4: Plots of the sample distribution (solid line), the predicted (discrete) distribution (dashed line), and the underlying continuous distribution for input variance $\sigma^2 = 2.78$.

imental observation and the theoretical prediction in that the variation of the sample mean is within six times the standard deviation of the predicted distribution. For input noise variances that are not too large, the quantization error is not severe, and we observe agreement between the theoretically predicted variance and the observed sample variance.

5 Conclusion

In this paper, we applied the theory of covariance propagation to template matching and obtained the variance of the output of the template matching algorithm as a function of the covariance of the input perturbation and the template itself. It is a measure of the discriminatory power of the template, and can be used to guide the construction of templates in order to increase the invariance of the templates and potentially increase the robustness of template matching algorithms to perturbations that do not closely follow the proposed stochastic model. Experiments on synthetic data agreed with the theoretical prediction. We are now conducting experiments on real data to test the performance of the methodology.

References

[Haralick, 1996] Robert M. Haralick, Propagating Covariance In Computer Vision, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 10, No. 5, 1996, pp. 561–572.

[Haralick, 1994] Robert M. Haralick, Performance Characterization Protocol in Computer Vision, *Proceedings of the 1994 IUW*, Monterey, CA, Nov. 1994. Vol. I, pp. 667–673.

[Liu and Haralick, 1997] Gang Liu, Robert M. Haralick, Performance Evaluation of Template Matching Approaches to Location Estimation (1-D case), technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle WA 98195, March 1997.

[Kanungo and Haralick, 1995] Tapas Kanungo, Robert M. Haralick, Multivariate Hypothesis Testing for Gaussian Data: Theory and Software, technical report, Intelligent Systems Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195, October 1995.

[Casella and Berger, 1990] George Casella, Roger L. Berger, *Statistical Inference*, Duxbury Press, California, USA, 1990.

[Press *et el.*, 1992] William H. Press, *et al.*, *Numerical Recipes in C : The Art of Scientific Computing*, 2nd Ed., Cambridge University Press, 1992.

Performance Modeling and Adaptive Target Detection

David M. Doria
Hughes Aircraft Company
P.O. Box 902, El Segundo Ca 90245
E0/E1/A172
doria@warp10.es.hac.com

Abstract

We present an approach to performance prediction of a geometric-based automatic target detection (ATD) system.[1] Performance analysis of both the probability of a false match to background and the probability of a match (detection) to the true target are presented. The predictive models lead to a means by which to automatically trade-off the expected probability of detection (Pd) and false alarm rate (FAR) for a given scene or part of a scene, thereby allowing a user-specified set of overall system performance specifications to drive the system parameters. Results using an adaptive algorithm are presented. It is seen in initial experimental results that the adaptive approach is able to significantly reduce the FAR at the expense of a relatively minor reduction in the Pd.

1 Introduction

An analysis of the matching method of fixing a maximum model-to-data local match tolerance and measuring the observed fraction of model matched to data is presented. To arrive at a tractable result, we have adopted a number of simplified models, including those for atmospheric attenuation, targets, correlation between target models, edge operators, sensor blur, sampling, noise, and background. We believe these approximations to be adequate for the present analysis, with a goal of functioning as a useable predictive system over a broad range of scenarios, targets, and sensors. Section 1 gives an overview of the analysis approach, which is followed in sections 2 and 3 by the FAR and Pd analyses, respectively. In section 4 we present results of initial experiments using an adaptive ATD system.

The false alarm analysis describes the probability of a false match within a local search area as a function of the local clutter edge density and correlation, search area, model set, and algorithm search and match parameters. The Pd is a function of the target contrast, size, range, the atmospheric attenuation, the sensor optical and detector point spread functions, the sensor sensitivity, the properties of the edge detector, and the fractional visibility of the target. Both the FAR and Pd performance models are functions of the fraction f_0 of the model matched to the data above which a detection is declared. This allows the FAR and Pd models to be expressed as simultaneous functions of f_0, and a predicted receiver operating characteristic (ROC) curve calculated. The ROC predictive capability enables trade-offs between the Pd and FAR to be performed by an adaptive ATD algorithm.

The ATD system discussed in this paper is based on the matching of geometric features of a model to those of extracted data. As such, the probability of extracting the true target versus background features, and the local tolerances of

[1] This work was sponsored by DARPA under Army Research Office contract TDAAH04-93-C-0052, and is currently sponsored under Air Force contract F33615-97-C-1022.

the model-to-data feature matches, tend to control the system Pd and FAR. Thus far we have only analyzed the use of target edge features. In the future we will extend this type of analysis to include other types of geometric information, and to optimize feature extraction with respect to the expected Pd and FAR [Doria 1997].

2 False Alarm Rate Prediction

We present an analysis of an operating mode in which the maximum distance of each local model-to-data edge match match is fixed, and the fraction of the model matched to the data is measured. Grimson and Huttenlocher [1994] present an analysis of the probability of false match for a generalized Hausdorff based matching algorithm over translation. They show expressions for the probability that a given model point will overlap the area within a fixed distance of any data point for non-correspondence based Hausdorff matching methods, and calculate subsequent false match probabilities. In [Doria 1996] the false match prediction model was extended to include the effects of correlated clutter edges, local model-to-data matching window size, correlation over translation, and search with multiple correlated models.

2.1 False Alarm Rate Calculation

In many ATR systems, areas of interest (AOIs) are obtained from an initial screening or focus of attention mechanism. These then restrict the pose search to a number of sub-areas of the image. If within any of these AOI's at least one model is accepted as a match when no target is present, we label this as a false alarm for that AOI. Only a single target report is allowed per AOI, so that multiple matches of different models to different locations within the AOI are not counted twice. The overall expected false alarm rate per image is then the sum of the mean false alarm probability at each AOI:

$$FAR_{\mathrm{Im}\,age} = \sum_{n_a}^{N_{AOI}} P_{FA}(n_a)$$

where n_a denotes a particular AOI, N_{AOI} is the number of AOI's, and $P_{FA}(n_a)$ is the probability of a false alarm in AOR n_a. Note that because instances of false matches are not counted more than once for each AOI, the value of $P_{FA}(n_a)$ (henceforward called simply P_{FA}) is bounded between 0 and 1. Using this terminology, we address the prediction of the false alarm probability P_{FA} under translational search for multiple models as a function of important parameters affecting the probability of a false match for a "maximum fraction at a fixed distance" Hausdorff based approach. The false alarm rate prediction is based on the combination of search space, number of models, complexity of the non-target regions, sensor characteristics, and range to the AOI's.

The FAR performance model addresses the performance of the system operating on geometric matches to edges. For this analysis we model the background as a zero mean noise process and analyze the performance of a simple edge operator of the form

$$g_h(i,j) = I(i,j-1) - I(i,j+1)$$
$$g_v(i,j) = I(i-1,j) - I(i+1,j)$$

with gradient magnitude

$$g(i,j) = \sqrt{g_h(i,j)^2 + g_v(i,j)^2}$$

Other more sophisticated edge operators can be be analyzed and inserted into the model as the user desires; we have chosen a relatively simple representative edge operator for the initial analysis.

The system operates independently in each AOI. We first estimate the expected density of background (or clutter) edges. Given zero mean clutter of variance σ_C^2, and covariance of σ_{C_2} at an offset of two pixels, the expected values of g_h and g_v are 0, and the variances of g_h and g_v are $\sigma_g^2 = 2\sigma_C^2 - 2\sigma_{C_2}$. The operator g then has a Rayleigh probability density over the background:

$$P(g(i,j)) = \frac{g(i,j)}{\sigma_g^2} e^{-\frac{g(i,j)^2}{2\sigma_g^2}}$$

To relate the P_{FA} to the actual background grey level statistics, we use an adaptively selected local edge detector threshold for the false alarm and detection performance models and within the algorithm. Given the above model of an edge operator[2], the probability of detecting a false edge in the background is $\rho_e = e^{\frac{-(\frac{t_e}{\sigma_g})^2}{2}}$.

where t_e is the edge operator threshold.

As a first order model of the correlation between background edges, we adopt an exponential function of distance for horizontal and vertical edges:

$$\rho_{x,\Delta x} = \rho^{|\Delta x|}$$

$$\rho_{y,\Delta y} = \rho^{|\Delta y|}$$

These, along with the clutter edge density ρ_e can be estimated from the detected clutter edges.

2.2 Matches at a Single Model Point

The Hausdorff based matching approach that we are studying operates by matching model features (in this case edges) to observed data. A distance h_d is used to allow or disallow model-to-data edge matches. If at least one data edge is within the distance h_d to a given model edge, a match is counted for the model edge. Use of a specified tolerance distance translates into the specification of a local window around each model edge point within which data edges can occur for an allowed match. For $h_d = \frac{1}{2}$ there is a corresponding 1×1window, and for $h_d = \sqrt{2}$ there is a 3×3 local window. We will focus on the performance of both 1×1 and 3×3 local windows. Alternatively, we can use an inverted distance transform, as described in [Doria and Huttenlocher 1996].

The probability of match of a single model point to noise data is a function of the density of noise (background clutter and noise) edges, the covariance of the background edges, and the size of the tolerance window surrounding each model point. First we will find the probability that a single model point m is matched to a data point. For a 1×1 tolerance window, the expected value of the sum of window points surrounding the model edge point m is

$$u_s = \rho_e$$

and for a 3×3 window the expected value of the sum is

$$u_s = 9\rho_e$$

where ρ_e is the estimated density of background noise or clutter edges. The variance of the sum over the window points is, when the pixels are independent,

$$\sigma_s^2 = N\rho_e(1-\rho_e)$$

where N is the number of pixels in the local window.

If the edge pixels in the window are not independent, then the variance of the sum is

$$\sigma_s^2 = \sum_{u=1}^{3}\sum_{v=1}^{3}\sum_{u'=1}^{3}\sum_{v'=1}^{3}\sigma(u,v,u',v')$$

where

$$\sigma(u,v,u',v') = \rho_x^{|u-u'|}\rho_y^{|v-v'|}\rho_e(1-\rho_e)$$

When the edge pixels are independent, the probability p_e of a match of a model point to at least one background data edge is

$$p_e = 1-(1-\rho_e)^N$$

[2] Note that other operators or feature types can be used; analysis of their performance in the presence of target and clutter would then be used instead of edges within this general modeling method. Application to other sensor types is also feasible.

In the case of correlated edges in a 3×3 (or larger) window, we can calculate the binomial statistics directly. Alternatively, we can estimate p_e using a Gaussian approximation for the sum:

$$p_e = 1 - \Phi(\frac{0.5 - u_s}{\sigma_s})$$

where Φ is the standard Gaussian distribution function. The approximation to a Gaussian density is not extremely accurate with only nine terms, especially with low values of u_s. As an alternative, we can estimate the equivalent number of "independent" pixels as $N' = \frac{N^2 \rho_e (1 - \rho_e)}{\sigma_s^2}$, and use a Poisson model:

$$p_e \approx 1 - e^{-\rho_e N' \frac{(\rho_e N')^0}{0!}}$$
$$= 1 - e^{-\rho_e N'}$$

Once we have estimated the probability of a match to a single model edge feature, we obtain the probability of a match to a fraction of the edge pixels of the entire model, at a single model location in clutter. The match of a model to data is based on the sum of model points that have a matching data point within a distance d_h of the model point.

Let $s_m = \sum_{n=1}^{N} x_{m,n}$ where the $x_{m,n}$ values are the binary edge values within the window surrounding model point m. We define a new binary random variable S_m as:

$$S_m = \begin{cases} 1 & if\ s_m \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

The sum of model matches over the entire model is then

$$S_M = \sum_{m=1}^{M} S_m$$

When S_M is greater than or equal to a threshold $f_0 M$, a detection is declared.

The statistic S_M is the sum of Bernoulli random variables, and can be approximated by a Gaussian distribution $N(u_B, \sigma_B^2)$. The expected value of the sum for the background pixels is $u_B = Mp_{e.}$. The variance of S_M is then the sum of the covariances of the model-to-data match windows:

$$\sigma_B^2 = \sum_{i=1}^{M} \sum_{j=1}^{M} \sigma_{S_{i,j}}$$

where *(i,j)* indicate model edge points. When the model-to-data matches are independent, the variance becomes the familiar binomial variance:

$$\sigma_B^2 = Mp_e(1 - p_e).$$

It is often the case that the local model to data match windows are not independent. This may be due to overlapping match windows around the model edge points, or statistical correlation between the data edges. For these cases we estimate the value of σ_B^2 from the above sum of covariances. Let $\rho_{S_{..,+k}}$ be the correlation between the random variables S_m and S_{m+k}. We use

$$\rho_{S_{m,m+k}} \sigma_S^2 = R_{S_{m,m+k}} - u_m^2$$

where for binary random variables

$$R_{S_{m,m+k}} = P(S_{m+k} = 1 \| S_m = 1) P(S_m = 1)$$
$$= P(S_{m+k} = 1 \| S_m = 1) p_e.$$

Letting $u_s' = E(s_m | s_m \geq 1)$, and $\rho_{m,m+k}$ be the correlation between s_m and s_{m+k} due to both statistical clutter correlation and geometric window effects, the conditional expected value $E(s_{m+k} | s_m \geq 1)$ is

$$E(s_{m+k} | s_m \geq 1) \approx \rho_{m.m+k}(u_s' - u_s) + u_s.$$
$$= u'_{s_{m,m+k}}$$

The conditional variance can also be estimated for each *(i,j)* combination of *m* and *m+k*.[3] From this, the variance σ_B^2 can be obtained.

A detection is determined by the number of model edges S_M that are matched to at least one data edge, as described above. The observation S_M is a sample from a binomial distribution, which is well modeled as a Gaussian distribution when M is large enough and S_M is not too near *O* or M. A detection is declared if $S_M \geq f_0 M$, and the corresponding probability of a false match $P_{fa}(f_0)$ at a single location of the model is approximated by

$$P_{fa}(f_0) \approx 1 - \frac{1}{\sqrt{2\pi}} \int_{d1_B}^{d2_B} e^{-\frac{z^2}{2}} dz$$

where $d1_B = \dfrac{-0.5 - u_B}{\sigma_B}$ and

$$d2_B = \frac{f_0 M - u_B - 0.5}{\sigma_B}$$

where f_0 has a value of $\dfrac{a}{M}, a \in 0,1,\ldots,M.$

2.3 Search Over an Area of Interest

As discussed earlier, we typically search each AOI to determine if a target exists within that area. For a single model, we now look at the probability of a false match within a search area of size *A*=*A*$_x$ × *A*$_y$. The probability density of the match values is expressed by the joint Gaussian density of the model matches as a function of translation over the search area *A*.

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{A}{2}} |\mathbf{C}|^{\frac{1}{2}}} e^{-\frac{-(\mathbf{x}-\mathbf{u_B})\mathbf{C}^{-1}(\mathbf{x}-\mathbf{u_B})}{2}}$$

where $\mathbf{C}$ is the covariance matrix between the model-to-data matches over translation, $\mathbf{x}$ is the vector of observed match scores over the area *A*, and $\mathbf{u_B}$ is the mean vector. The covariance matrix $\mathbf{C}$ is a function of the model size and shape, and the local model-to-data tolerance window.

For this analysis, we represent models as rectangles of dimensions $L_x \times L_y$. As an element of the estimation of the FAR, we need to estimate the correlation between the model-to-data scores as a function of model horizontal and vertical translations Δx and Δy. This requires an estimate of the correlation of the model itself as a function of translation. We approximate the model correlations $\rho_{M,x}$ and $\rho_{M,y}$ as:

$$\rho_{M,x} \approx \frac{L_x - \Delta x}{(L_x + L_y) - 2} \delta(\Delta y)$$

$$\rho_{M,y} \approx \frac{L_y - \Delta y}{(L_x + L_y) - 2} \delta(\Delta x)$$

A single correlation value is obtained from

$$\rho_{M,xy} = \sqrt{\rho_{M,x} \rho_{M,y}}$$

2.4 Probability of One or More False Matches per Area

We are interested in the probability that at least one of the matches over translation exceeds a fraction f_0 matched to the data. To find this we express the matches at each location as a product of conditional probabilities, and then estimate the probability that at least one of the matches exceeds a threshold. Denoting the model match results at locations $1,2,\ldots,A$ as $\mathbf{x}$ = [*x*$_1$, *x*$_2$, …,*x*$_A$,]^T, the probability of false alarm over the area *A*, at a match fraction threshold of f_0, is:

$$P_{FA}(A, f_0) = 1 - P(x_1 < M_{f_0}) P(x_2 < M_{f_0} | x_1 < M_{f_0})$$
$$\ldots P(x_A < M_{f_0} | x_{A-1} < M_{f_0}, x_{A-2} < M_{f_0}, \ldots,$$
$$x_1 < M_{f_0})$$

The conditional mean of a Gaussian random variable $\mathbf{x}_2 = x_i$, given observed variables $\mathbf{x}_1 = [x_1, x_2, \ldots, x_{i-1}]$ is

[3] See [Doria 1996] for details.

$$\overline{\mathbf{x}}_{2|1} = \overline{\mathbf{x}}_2 + \mathbf{C}_{21}\mathbf{C}^{-1}{}_{11}(\mathbf{x}_1 - \overline{\mathbf{x}}_1)$$

where the covariance matrix has been partitioned as:

$$\mathbf{C}_{2|1} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix}$$

and conditional variance is

$$\mathbf{C}_{2|1} = \mathbf{C}_{22} - \mathbf{C}_{21}\mathbf{C}^{-1}{}_{11}\mathbf{C}_{12}.$$

Since we expect that the conditional mean, for meaningfully low false alarm rates, will be approximately equal to the unconditional mean, we let $\overline{\mathbf{x}}_{2|1} = u_B$. If we model the search between rows as independent, and treat correlation as a function of model translation along a row as Markov, then the conditional variance is

$$\sigma_B^{2'} = (1 - \rho_{M,xy}^2)\sigma_B^2$$

The expression for $P_{FA}(A, f_0)$ now becomes:

$$P_{FA}(A, f_0) \approx 1 - \frac{1}{(2\pi)^{\frac{A}{2}}} \left(\int_{d1_A}^{d2_A} e^{\frac{-z^2}{2}} dz \right)^{A_y} \left(\int_{d1'_A}^{d2'_A} e^{\frac{-z^2}{2}} dz \right)^{A_y(A_x - 1)}$$

where

$$d2_A = \frac{f_0 M - u_B - 0.5}{\sigma_B}, d1_A = \frac{-0.5 - u_B}{\sigma_B},$$

$$d2'_A = \frac{f_0 M - u_B - 0.5}{\sigma'_B}, \text{ and } d1'_A = \frac{-0.5 - u_B}{\sigma'_B}.$$

2.5 Search With Multiple Models

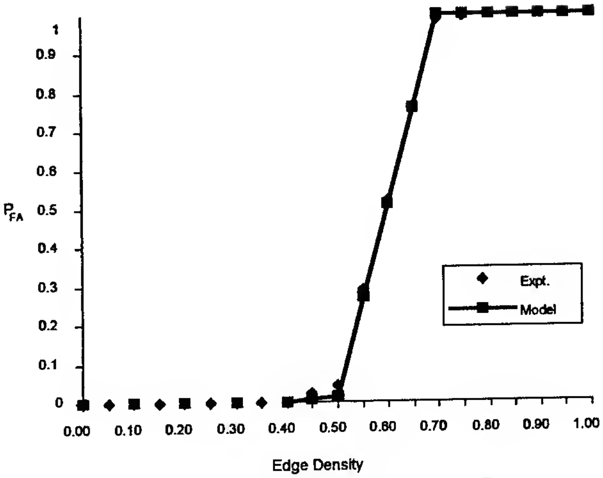


Figure 1. Probability of false alarm P_{FA} as a function of edge density ρ_e, for both predicted and Monte-Carlo experimental results; 5×5 search area, single 6m $\times$ 3m model, 1×1 Hausdorff matching window, f_0 =.75, R=6km.

In most real ATR problems the system must search an area for more than a single target type (or type,view combination). In such instances the false alarm probability within a given area is a function of the number of models and their correlation. In this section, we approximate the statistical correlation of model-to-data matches between models, and modify the expression for P_{FA} accordingly. To precisely model the P_{FA} performance requires that the covariance matrix between the model types (or types $\times$ views) be known. For a general prediction capability, however, a model of approximate behavior for a general set of models is more useful. To achieve this, we study the case of a set of models of equal size and equal correlation, so that the covariance matrix between the models is:

$$C_M = \sigma_B^2 \begin{bmatrix} 1 & \rho_M & \cdots & \rho_M \\ \rho_M & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \rho_M \\ \rho_M & \cdots & \rho_M & 1 \end{bmatrix}$$

where all non-diagonal elements of the correlation matrix are $\rho_M = \overline{\rho}_m$, where $\overline{\rho}_m$ is the mean between-model normalized cross correlation.[4]

[4] Clearly this is not descriptive of most actual model sets, but is a reasonable first order approximation. Other model set correlations can also be adopted if desired, and actual model-set correlations can be used if known.

We treat the search over multiple models as an independent search dimension at each translational location *t*. We now have, for N_T models, a joint distribution of matches of N_T models at model location *t*. Expressing the probability of a false match as a function of conditional probabilities, we have

$$P_{FA}(f_0, M, N_T, t) = 1 - P(x_{1,t} < f_0 M) \cdot$$
$$P(x_{2,t} < f_0 M | x_{1,t} < f_0 M) \ldots P(x_{N_T,t} < f_0 M |$$
$$x_{1,t} < f_0 M, x_{2,t} < f_0 M, \ldots, x_{N_T - 1,t} < f_0 M)$$

By assuming joint Gaussian statistics for the above conditional probability densities for the set of models at location *t*, we can calculate conditional means and variances, and express the above equation as a product of conditional probabilities of each successive model. We again set the conditional mean to the value of the unconditional mean, and the conditional variance of model m to

$$\mathbf{C}_{2|1,m} = \mathbf{C}_{22,m} - \mathbf{C}_{21,m}\mathbf{C}_{11,m}^{-1}\mathbf{C}_{12,m}.$$

For the above covariance matrix this gives conditional variances of either

$$\sigma_{B,m}^2 = \sigma_B^2 \left(1 - \frac{(m-1)\rho_M^2}{(m-2)\rho_m + 1} \right)$$
$$\sigma_{B,m}^2{}' = \sigma_B^2{}' \left(1 - \frac{(m-1)\rho_M^2}{(m-2)\rho_m + 1} \right).$$

We now have an expression for the probability of false match for translation over a search area *A*, accounting for correlated background clutter, correlated local match windows, correlation between models, and correlation of models as a function of translational location:

$$P_{FA}(A_x, A_y, f_0, M, N_T, \rho_M) = 1 -$$
$$\frac{1}{(2\pi)^{\frac{A_x A_y N_T}{2}}} \left(\prod_{m=1}^{N_T} \int_{\frac{-.5-u_B}{\sigma_{B,m}}}^{\frac{f_0 M - u_B - .5}{\sigma_{B,m}}} e^{-z^2/2} \right)^{A_y} \cdot$$
$$\left(\prod_{m=1}^{N_T} \int_{\frac{-.5-u_B}{\sigma_{B,m}'}}^{\frac{f_0 M - u_B - .5}{\sigma_{B,m}'}} e^{-z^2/2} \right)^{A_y(A_x - 1)}$$

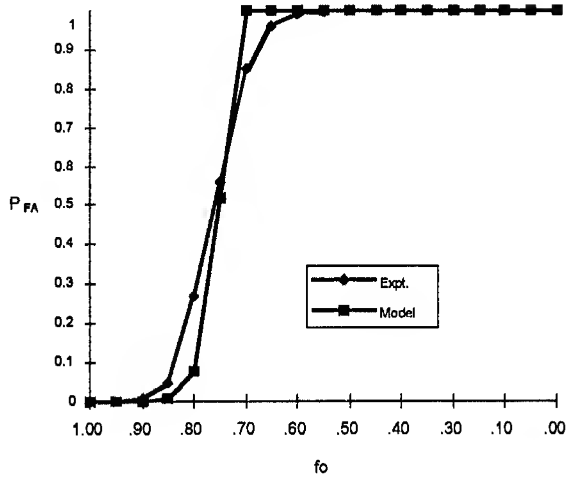


Figure 2. Probability of false alarm P_{FA} as a function of model match threshold f_0 for 10 equally correlated models (ρ_M =.75) of 18m contour, for both predicted and Monte-Carlo experimental results; 5×5 search area, 3×3 matching window, edge density ρ_e =.1, R=3km.

3 Probability of Detection Modeling

3.1 Pd Analysis

In this section we model the performance of the Hausdorff based geometric matching algorithm on true targets, again using rectangular target models of dimensions $L_x \times L_y$. Let the temperature difference between the target and background be ΔT at range *R*. We model the atmospheric attenuation as exponential to obtain an effective temperature difference

$\Delta T_R = \Delta T e^{-\beta_{Atm} R}$ at the sensor. We adopt a first order model of a FLIR sensor as a linear system blur followed by sampling plus additive noise. The image irradiance function at the focal plane is then

$$H(i,j) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} d(x,y) h_{optics}(i-x, j-y)dxdy$$

where $d(x,y)$ is the image irradiance at the input to the sensor. The irradiance function is then convolved with the detector sampling function $h_{\det}$ and noise of variance σ_{El}^2 added:

$$I(i,j) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(x,y) h_{\det}(i-x, j-y)dxdy + w(i,j)$$

Normalizing for both the noise equivalent temperature difference ($NE\Delta T$) and the edge detector difference operation gives a unit variance in both $g_h(i,j)$ and $g_v(i,j)$ when operating on uniform areas of the target or background:

$$I(i,j)' = \frac{I(i,j)}{\sqrt{2}NE\Delta T}$$

and

$$\Delta T_R' = \frac{\Delta T_R}{\sqrt{2}NE\Delta T} \quad .$$

Given a target edge of effective thermal contrast $\Delta T_R'$, we wish to estimate the output of the edge operator discussed previously. To do this, we approximate h_{optics} as resulting from a circular aperture with diffraction limited blur, using the mean wavelength $\overline{\lambda}$ observed by the sensor, and the effective sensor optical diameter D. Alternatively, we can use a measured or estimated point spread function. We approximate the detector sampling blur kernal $h_{\det}$ as a two dimensional *Rect* function of size $IFOV_h \times IFOV_v$, and the pixel dimensions of size $P_{h_{FOV}} = \frac{IFOV_h}{P_{h_{samp}}}$ in h and $P_{v_{FOV}} = \frac{IFOV_v}{P_{v_{samp}}}$ in v, where $P_{h_{samp}}$ and $P_{v_{samp}}$ are the respective number of pixel samples per detector instantaneous fields of view in the h and v directions. The expected value of the output of the edge operator, with a vertical edge centered at location *(i,j),* is calculated from the expected edge operator h and v responses:

$$E(g_v(i,j)) \approx \Delta T_R' \left(\int_{-P_{h_{FOV}} - \frac{IFOV_h}{2}}^{-P_{h_{FOV}} + \frac{IFOV_h}{2}} H(i+u, j)du - \int_{P_{h_{FOV}} - \frac{IFOV_h}{2}}^{P_{h_{FOV}} + \frac{IFOV_h}{2}} H(i+u, j)du \right)$$

$$= \overline{g}_v$$

and

$$\overline{g}_h = E(g_h(i,j)) = 0$$

Letting $z = \sqrt{g_h^2 + g_v^2}$ (dropping the indices *(i,j)*), the density function of the edge operator at the center of either a vertical or horizontal edge transition is now Rician[5]. Using the normalized values of the thermal contrast, we have unit variance in g_h and g_v, and a distribution of the edge values of

$$P(z) = ze^{-\frac{z^2+\overline{g}_v^2}{2}} I_0(z\overline{g}_v)$$

The probability of detection of a single target edge is then

$$\rho_{T_e} = \int_{t_e}^{\infty} ze^{-\frac{z^2+\overline{g}_v^2}{2}} I_0(z\overline{g}_v)$$

which is Marcum's Q-function.

[5] We have not included "off-edge" true responses.

Given ρ_{T_e}, we wish to estimate the probability that a given model edge will be matched to an observed data edge from the target[6]. For a 1×1 matching window, the probability of a match is

$$p_{T_e} = \rho_{T_e}$$

For a 3×3 matching window, the target contour passes through approximately three edge pixels in the window; hence the probability of a model-to-data edge match is

$$p_{T_e} \approx 1 - (1 - \rho_{T_e})^3$$

For a target of contour length $L_T = L_x L$ meters, the number of observed pixels is:

$$M_T \approx \frac{2L_x P_{h_{samp}}}{(R)(IFOV_h)} + \frac{2L_y P_{v_{samp}}}{(R)(IFOV_v)} - 4$$

The distribution of observed target pixels matched to the correct target model is binomial with mean

$$u_T = p_{T_e} M_T$$

and variance

$$\sigma_T^2 = M_T p_{T_e} (1 - p_{T_e})$$

For correlated tolerance windows, we also adjust the variance of the model match sum similarly to that with the false alarm, arriving at a variance $\sigma_T^{2'}$. We now estimate the probability of detection of the target by approximating the binomial distribution with a Gaussian:

$$P_d(\Delta T_R, f_0, L_T . R, \beta_{Atm}, NE\Delta T, IFOV_h, IFOV_v, h_{optics}, h_d) =$$

$$1 - \frac{1}{\sqrt{2\pi}} \int_{\frac{-.5-u_T}{\sigma_T'}}^{\frac{f_0 M - .5 - u_T}{\sigma_T'}} e^{-\frac{z^2}{2}} dz$$

A few examples indicate the predictive use of this equation[7]. Figure 3 shows Pd as a function of the matching fraction f_0 and a 1×1 model matching tolerance window. Figure 4 shows the same experiment with a 3×3 tolerance window.

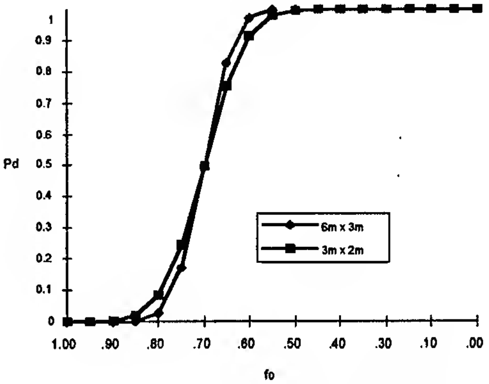


Figure 3. Predicted probability of detection as a function of detection threshold f for a 6m × 3m target and a 3m × 2m target. $\rho_{T_e} = .7$, 1×1 tolerance window, range = 3km, $IFOV = 75u$ rad.

3.2 Partially Observable Target

It is often the case that a target may be only partially visible. This may be the result of either very low thermal contrast over a portion of the target, partial occlusion by a portion of the surroundings, or intentional concealment. In such cases the maximum observable number of target pixels is $M_T'' = (1 - f_{occl}) M_T$, where the fractional non-visibility is f_{occl}. In such cases the equation for the predicted Pd becomes

[6] We assume that the target model is positioned at the correct location (i.e. we have not analyzed the search process here; we assume that a model is positioned, by whatever means, at the correct location).

[7] The sensor parameters used in the examples in this paper were not derived from nor do they represent any actual or potential sensor parameters. Any similarity between these parameters and those of an actual sensor is coincidental.

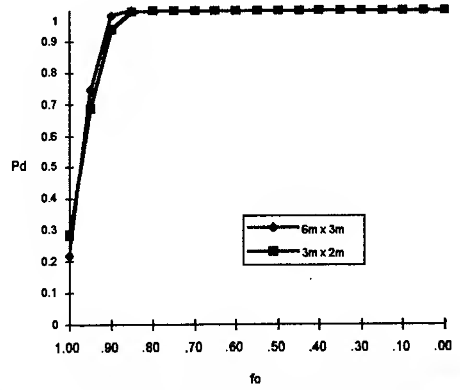


Figure 4. Predicted probability of detection P_d as a function of detection threshold f_0 for a 6m × 3m target and a 3m × 2m target. Same parameters as in Figure 3 except now the system is using a 3×3 tolerance window for matching each model point to an edge point.

$$P_d(\Delta T_R, f_0, L_T. R, \beta_{Atm}, NE\Delta T, IFOV_h, IFOV_v, h_{optics}, h_d, f_{occl}) =$$

$$\begin{cases} 1-\frac{1}{\sqrt{2\pi}}\left(\int_{\frac{-.5-u_T''}{\sigma_T''}}^{\frac{f_0 M_T - u_T'' -.5}{\sigma_T''}} e^{-\frac{z^2}{2}} dz + \int_{\frac{M_T'' - u_T'' -.5}{\sigma_T''}}^{\frac{M_T - u_T'' -.5}{\sigma_T''}} e^{-\frac{z^2}{2}} dz \right) & \text{if} f_0' < 1 - f_{occl} \\ 0 & \text{otherwise} \end{cases}$$

To obtain an equivalent probability of detection in the presence of a partially visible target, the threshold f_0 must be reduced. This increases the expected *FAR*, and reduces the separability of target and clutter.

3.3 ROC Calculation

By obtaining predictions of both the Pd and P_{FA} as a function of f_0, predicted receiver operating characteristic (ROC) curves are able to be calculated by the system. This can be done for each local area of interest, using the estimated clutter characteristics from that area. An example of an ROC curve predicted using the Pd and P_{FA} models is given in Figure 5.

4 Adaptive Detection System

Performance specifications for ATD/R systems are typically expressed in terms of minimum Pd and maximum FAR levels. The relationship between these specifications and actual algorithm performance has in the past, for the most part, been determined empirically for each set of training data. Test data that deviated from the training data often has resulted in unpredictable and uncontrollable performance using ATD/R parameters trained or tuned for a particular database. This is generally true for both statistical pattern recognition and model based approaches, although model based approaches are usually easier to re-tune. The introduction of predictive relationships between the algorithm match quality statistic and the Pd and FAR enables a mechanism whereby the adaptive ATD/R system can calculate a range of model fit values that are within Pd and FAR specifications set by the user.

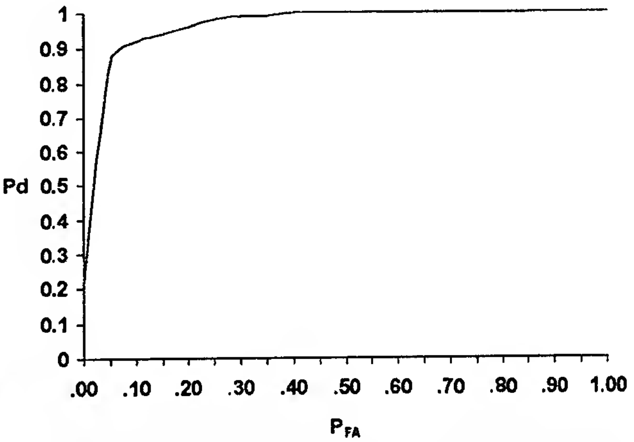


Figure 5. Predicted probability of detection Pd as a function of P_{FA} for a set of 10 6m × 3m models with $\rho_M = .75$, target dimensions of 6m × 3m, R=3km, $IFOV = 75\, urad$, target $\rho_{T_e} = .7$, 5×5 pixel search area, background edge density $\rho_e = .14$, and a 3×3 Hausdorff tolerance window.

In an initial set of experiments, we used a test statistic $y = \frac{d_M(\mathbf{x})}{P_{FA}(f_0)}$, where $d_M(\mathbf{x}) = \sum_{m=1}^{M} f_d(\mathbf{x_m})$, and $f_d(\mathbf{x_m})$ is the value of a clipped inverted distance transform [Doria and Huttenlocher 1996].

In the future we will also be running experiments with a test statistic defined as $y = \frac{P_d(f_0^{'})}{P_{FA}(f_0^{'})}$, where $f_0^{'}$ is an observed best fraction of model matched to data. Results from initial tests on sample FLIR target data from the UGV RSTA program are shown in Figure 6. For these detection level tests, a screener was developed that combines use of both thermal contrast and geometric information to improve initial target detection. The number of screener AOI's was set to a maximum number of expected targets in the scene (in this case 6). Following the screening function, potential target areas of interest were passed on to the matching (detection level) system. Both adaptive and non-adapative detection level match statistics were generated. As seen in the figure, a large fraction of the false alarms were eliminated by the adaptive system at the expense of a small loss in probability of detection. These results indicate the potential value of performing this type of adaptive analysis and processing as a component of model based automatic target detection, recognition, and image exploitation systems.

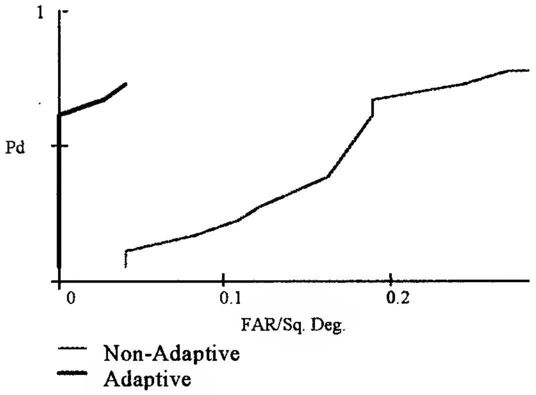


Figure 6. Example of adaptive and non-adaptive processing on a set of 10 frames from the RSTA database.

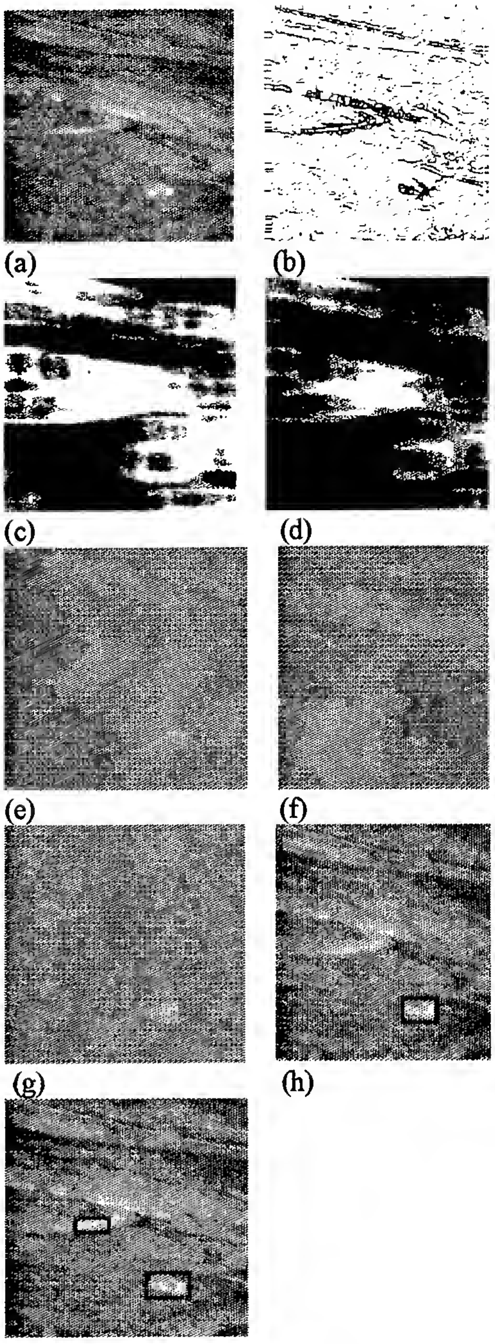


Figure 7. Results of adaptive algorithm on sample RSTA FLIR image: (a) original image, (b) edge image, (c) P_{FA} image at f_o = .80 (white indicates high probability of a false alarm in the local area) (d)) P_{FA} image at f_o = .95, (e) detection evidence, (f) adaptive detection statistic at f_o = .80, (g) adaptive detection statistic at f_o = .95, (h) detections at detection statistic f_o = .80, (i) detections at detection statistic f_o = .95. System had Pd = 1.0 and 1 FA/image at f_o = .95, and Pd = 1.0 and 0 FA/image at f_o = .80.

5 Summary

We have presented a performance model for an automatic target detection system that operates on model information contained in geometric features. The extraction of edge based geometric information is related to target thermal contrast in the present analysis. The false alarm rate is seen to be a function of the local search area, the clutter density and spatial correlation, the number of target models used in the detection-level search process, the size of the models, the correlation between models, the sensor sampling characteristics, the algorithm match quality tolerance, and the maximum fraction of a model matched to the data. The probability of detection is given as a function of the effective target temperature, atmospheric attenuation, range-to-target, the sensor optics, sampling, and sensitivity, the model size, the match tolerance window size, and the expected fraction of the model features matched to data features. We will continue to evalueate the generalizing assumptions used in the present development with respect to their extensibility and accuracy for performance prediction on a wide variety of scenarios. By combining the P_{FA} and Pd models, expected receiver operating characteristic (ROC) curves are calculable using this approach. The fact that terms arising from the important factors controlling both the Pd and FAR allows trade-offs with parameters describing each term, and contributes to a general understanding of the relationship between these parameters in terms of the basic ATR performance metrics of probability of detection and false alarm rate for this type of algorithm. Trades between Pd and FAR are also now possible for different sensors and scenarios.

This type of algorithm performance model can also be extended to predict recognition and identification capability, as discussed in [Doria 1997]. It is likely that the performance of other types of automatic target detection and recognition systems that rely on combined shape and signature information, and that use local features, is also related in a general way to the development presented here.

References

[Grimson and Huttenlocher 1994] W. E. L. Grimson and D. P. Huttenlocher. "Analyzing the Probability of a False Alarm for the Hausdorff Distance Under Translation," Proc. ARPA Image Und. Wrkshp., Nov 13-16, 1994, 1257-1262.

[Huttenlocher et al. 1993] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. "Comparing Images Using the Hausdorff Distance," IEEE Trans. Patt Anal. and Mach. Intell., 15(9),850-863,1993.

[Castleman 1979] K. Castleman,``Digital Image Processing," Prentice Hall, 1979.

[Doria and Huttenlocher 1996] D. M. Doria and D. P. Huttenlocher, "Progress on the Fast Adaptive Target Detection Program," Proc. 1996 DARPA IUW, V. 1, 589, Palm Springs, Ca, 12-15 Feb., 1996.

[Doria 1996] D.M. Doria, "Geometric Model-Based FLIR ATR Performance Prediction," Proc. 5th. ATR Science and Technology Symposium, Johns Hopkins APL, July 1996.

[Doria 1997] D.M. Doria, "Image Understanding at Hughes Aircraft Company: Adaptive Image Exploitation," This proceedings.

Linear Models for Infrared Spectra

Glenn Healey and Luis Benites

Computer Vision Laboratory
Electrical and Computer Engineering
University of California
Irvine, CA 92697
healey,lbenites@ece.uci.edu
http://www.cvl.uci.edu

Abstract

We analyze the use of linear models for infrared (IR) spectral reflectance functions. These models have been studied extensively for visible wavelengths and form the basis of several approaches to estimating surface properties from color images. The infrared analysis is performed using a set of measured spectral reflectance data over the mid-wave and long-wave regions of the IR spectrum. The results show that low-dimension linear models provide an accurate approximation to a large collection of natural and manmade materials.

1 Introduction

The availability of multispectral and hyperspectral infrared imagers provides the opportunity to develop image understanding systems with capabilities far exceeding those of systems using visible wavelength sensors. Several multiband infrared sensors have been demonstrated spanning various wavelength ranges. These include HYDICE for the near-IR (210 bands over $0.4 - 2.5\mu m$), ARES for the mid-wave IR (75 bands over 2.0-6.3 μm), and AES for the long-wave IR (26,000 bands over $2.3 - 15.4\mu m$). Data from these sensors can be exploited by systems to achieve tasks such as object recognition and terrain classification in various environments.

The large number of bands captured by hyperspectral infrared sensors presents a challenge to modules which are required to represent and process this data. For visible wavelengths, low-dimensional linear models for spectral reflectance have been used to reduce the dimensionality of spectral representations. The use of these models is justified by several studies [1] [5] [7] which show that visible spectral reflectance functions can be approximated accurately using a linear combination of a few fixed basis functions. These models exploit the structure inherent in spectral reflectance functions to improve representational efficiency for many applications [6] [11].

In addition to providing representational efficiency, linear models also play an important role in several color constancy algorithms which are designed to recover illumination-invariant surface information from multispectral images [3]. In general, color constancy is an underconstrained problem for which the structure introduced by linear reflectance models permits a solution. The use of linear models also allows explicit characterization of the set of surfaces for which an algorithm will recover illumination-invariant surface descriptors. Several methods derived from linear models have been used for recognition in complex scenes with uncontrolled illumination [2] [4] [10].

In this paper, we examine the use of low-dimensional linear models for infrared reflectance functions. The study uses measured reflectance data for 88 natural and manmade materials. For the analysis, the infrared wavelengths are partitioned into the mid-wave spectral window of the atmosphere (MWIR) from $3\mu m - 5\mu m$ and the long-wave spectral window of the atmosphere (LWIR) from $8\mu m - 12.5\mu m$. The results indicate that infrared spectral reflectance functions and spectral emissivity functions can be represented accurately using linear models with a small number of parameters.

2 Linear Reflectance Models

Consider a set of M materials with spectral reflectance functions $s_1(\lambda), s_2(\lambda), \ldots, s_M(\lambda)$. An n-dimensional linear reflectance model for these materials is defined by a set of n basis functions $S_1(\lambda), S_2(\lambda), \ldots, S_n(\lambda)$ where each reflectance function is approximated by

$$s_i(\lambda) \approx \sum_{1 \leq j \leq n} \sigma_{ij} S_j(\lambda) \tag{1}$$

Thus, the linear reflectance model allows each reflectance function $s_i(\lambda)$ to be represented using the n weights $\sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{in}$. For systems starting from measured data, the spectral reflectance functions $s_i(\lambda)$ and the basis functions $S_j(\lambda)$ are represented at a set of W discrete wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_W$. The error in the approximation for a single reflectance $s_i(\lambda)$ is given by

$$E_i = \sum_{1 \leq k \leq W} (s_i(\lambda_k) - \sum_{1 \leq j \leq n} \sigma_{ij} S_j(\lambda_k))^2 \tag{2}$$

The basis functions $S_j(\lambda)$ are typically selected to minimize the total square error

$$E_T = \sum_{1 \leq i \leq M} E_i \tag{3}$$

using a procedure based on the singular value decomposition. The important question is how large n must be to represent accurately the set of reflectance functions.

3 IR Reflectance Data Analysis

Measured infrared spectral reflectance data for 45 manmade and 43 natural materials was used for the analysis. The data for natural materials was obtained from the Remote Sensing Laboratory at Johns Hopkins University and includes measurements for various minerals, vegetation, rocks, and soils. The measurement techniques are described in [8] and [9]. The data for manmade materials was obtained from the National Photographic Interpretation Center and includes measurements for concrete, road asphalts and tar, construction materials, paints, and roofing materials. From Kirchhoff's law, the spectral emissivity $\epsilon(\lambda)$ for opaque materials is related to the spectral reflectance $s(\lambda)$ by $\epsilon(\lambda) = 1 - s(\lambda)$ so that a linear reflectance model captures spectral emissivity as well. The analysis was performed separately for the mid-wave ($3\mu m - 5\mu m$) and long-wave ($8\mu m - 12.5\mu m$) spectral windows of the atmosphere.

3.1 Analysis for Mid-Wave IR

Eighty-seven of the materials were considered over $3\mu m - 5\mu m$ using 101 wavelength samples separated by $0.02\mu m$. (One of the 88 original materials did not have measurements for these wavelengths.) Before fitting the model, each sampled reflectance function $s(\lambda_k)$ was scaled by a constant so that

$$\sum_k (s(\lambda_k))^2 = 1 \tag{4}$$

This normalization ensured that each reflectance function was considered equally by the fitting process. The SVD was then used to find the basis functions $S_j(\lambda)$ which minimize E_T in (3). The first five basis functions are plotted in figure 1 and the average error $E_T/87$ for models using between 2 and 5 basis functions is shown in figure 2. The error bars in figure 2 indicate the standard deviation of the E_i values for each n. Figure 3 and figure 4 show the individual fits with the largest error using four and five basis functions. Figure 5 shows the individual fit with the smallest error using two basis functions.

We also analyzed the sets of 44 manmade and 43 natural materials separately. Figure 6 shows the dependence of the average error on the number of basis functions for the sets of natural and manmade materials. For each value of n, the collection of natural materials has the smaller error. This suggests that the natural materials have a higher degree of correlation with wavelength than the manmade materials.

3.2 Analysis for Long-Wave IR

All eighty-eight materials were considered over $8 - 12.5\mu m$ using 46 wavelength samples separated by $0.1\mu m$. As for the mid-wave IR, each reflectance function was normalized to have unit power as in (4) and the SVD was used to find the basis functions which minimized E_T. The first five basis functions are plotted in figure 7 and the average error $E_T/88$ for models using between 2 and 5 basis functions is shown in figure 8. The error bars in figure 8 indicate the standard deviation of the E_i values for each n. Figures 9 and 10 show the individual fits with the largest error using four and five basis functions. Figure 11 shows the individual fit with the smallest error using two basis functions. Figure 12 shows that for the long-wave IR, the linear model error is also larger for manmade materials than for natural materials.

References

[1] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychonomic Sci.*, 1:369, 1964.

[2] G. Healey and A. Jain. Retrieving multispectral satellite images using physics-based invariant representations. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8):842–848, August 1996.

[3] G. Healey and Q.-T. Luong. Color in computer vision: recent progress. In C.H. Chen, L.F. Pau, and P.S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*. World scientific, 1997.

[4] G. Healey and L. Wang. Illumination-invariant recognition of texture in color images. *J.Opt. Soc. Am. A*, 12(9):1877–1883, September 1995.

[5] L. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10):1673–1683, October 1986.

[6] D. Marimont and B. Wandell. Linear models of surface and illuminant spectra. *J. Opt. Soc. Am. A*, 9(11):1905–1913, November 1992.

[7] J.P.S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of the Optical Society of America A*, 6:318–322, 1989.

[8] J. Salisbury and D. D'Aria. Emissivity of terrestrial materials in the 3-5 μm atmospheric window. *Remote Sensing of Environment*, 47:345–361, 1994.

[9] J. Salisbury, A. Wald, and D. D'Aria. Thermal-infrared remote sensing and Kirchhoff's law, 1. laboratory measurements. *Journal of Geophysical Research*, 99:11,897–11,911, 1994.

[10] D. Slater and G. Healey. The illumination-invariant recognition of 3D objects using local color invariants. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(2):206–210, February 1996.

[11] B. Wandell. The synthesis and analysis of color images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-9(1), January 1987.

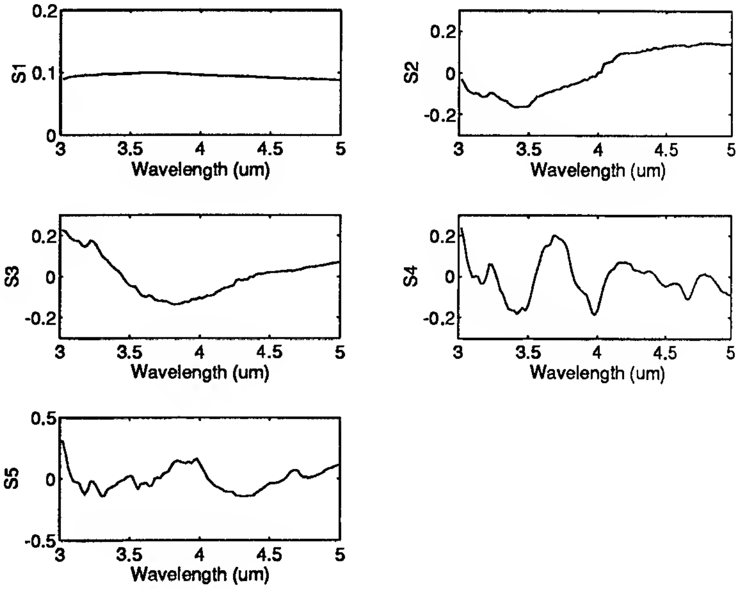


Figure 1: Basis functions for $3\mu m - 5\mu m$

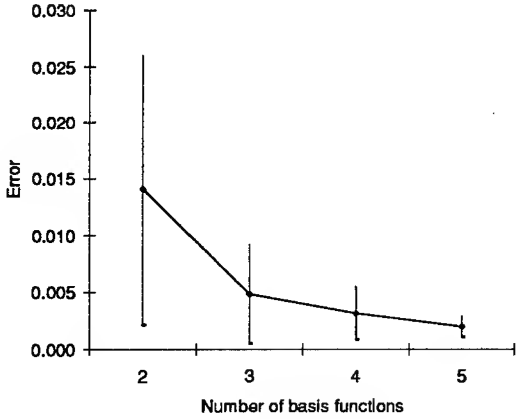


Figure 2: Average error for $3\mu m - 5\mu m$

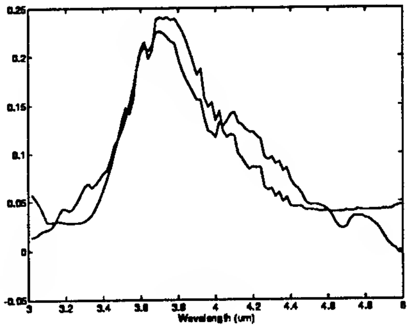


Figure 3: Worst fit for $n = 4$ for $3\mu m - 5\mu m$

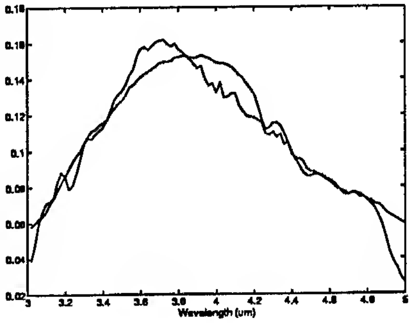


Figure 4: Worst fit for $n = 5$ for $3\mu m - 5\mu m$

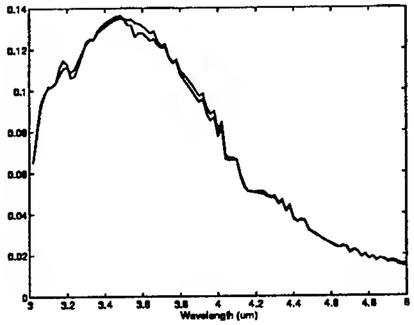


Figure 5: Best fit for $n = 2$ for $3\mu m - 5\mu m$

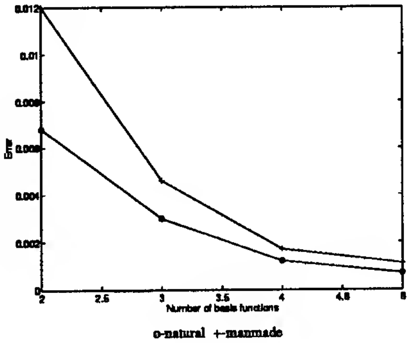


Figure 6: Error for natural and manmade materials

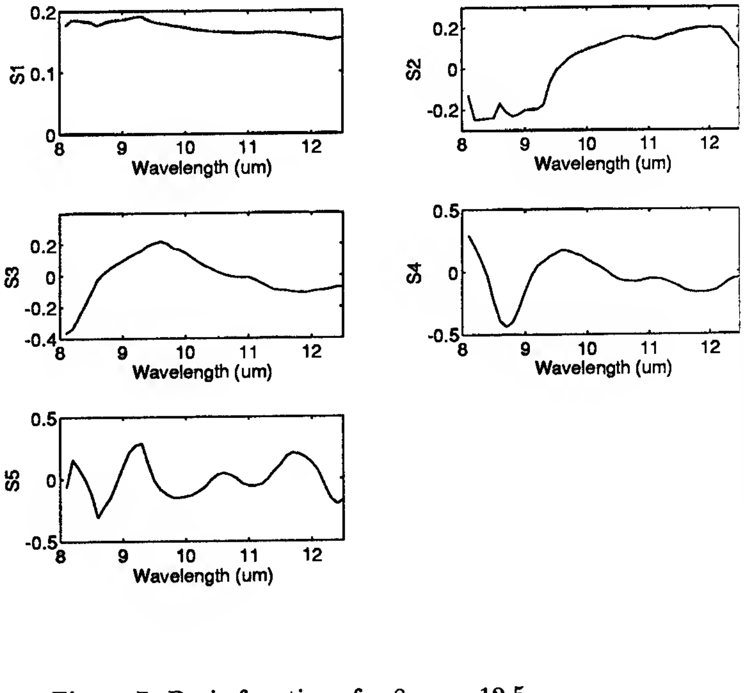


Figure 7: Basis functions for $8\mu m - 12.5\mu m$

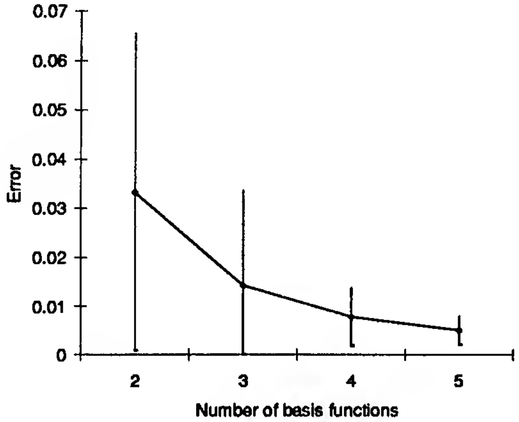


Figure 8: Average error for $8\mu m - 12.5\mu m$

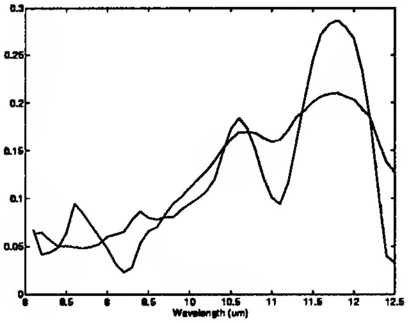


Figure 9: Worst fit for $n = 4$ for $8\mu m - 12.5\mu m$

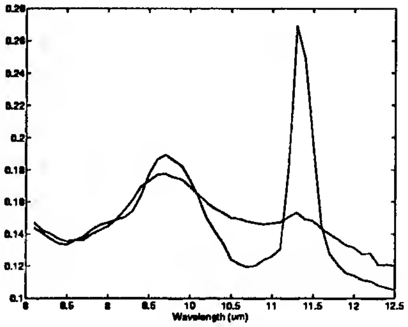


Figure 10: Worst fit for $n = 5$ for $8\mu m - 12.5\mu m$

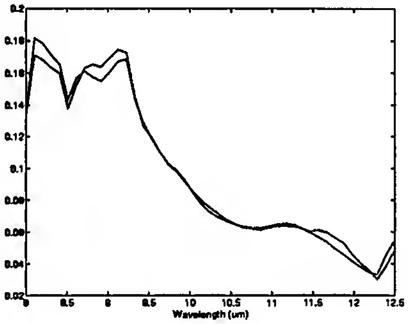


Figure 11: Best fit for $n = 2$ for $8\mu m - 12.5\mu m$

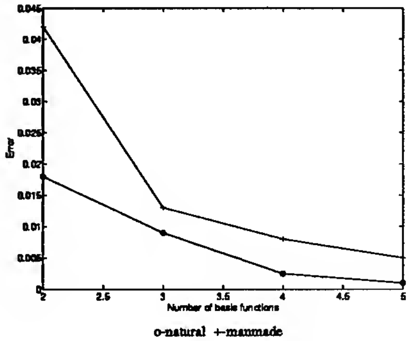


Figure 12: Error for natural and manmade materials

Recognition Using Multiband Filtered Energy Matrices

Lizhi Wang and Glenn Healey

Computer Vision Laboratory
Electrical and Computer Engineering
University of California
Irvine, CA 92697
liz,healey@ece.uci.edu
http://www.cvl.uci.edu

Abstract

We present an energy matrix representation for multiband images which captures spatial and spectral properties. Using a physical model for spectral reflectance, we derive a pseudoinverse method for the comparison of energy matrices which is invariant to the spectral properties of the scene illumination. At the same time, this method determines the illumination change matrix allowing direct comparison of images obtained under different illumination conditions. We demonstrate the performance of the method for both illumination-invariant recognition and illumination correction on a large set of multiband images. The energy matrices are generated using a small set of oriented steerable filters. We also demonstrate that a related set of rotationally symmetric filters can be used for recognition invariant to both illumination and rotation and that subsequent processing can be used to recover the rotation angle of a recognized object.

1 Introduction

Features extracted from the output of a set of spatial filters are often used for image representation [2] [9] [11]. Depending on the characteristics of the filters, these features can encode a large amount of information about orientation, scale, and spatial structure in an image. The application of spatial filters to multiband images enables the construction of feature sets which capture a wide range of spectral and spatial properties. These properties can be selected to optimize performance criteria for specific applications.

Despite these advantages, filter-based features are often not directly useful for indexing during recognition. Changes in the spectral properties of the scene illumination, for example, can lead to significant changes in filter-based features for a fixed object. This dependence on illumination conditions limits the usefulness of filter-based features for recognition to environments where the illumination can be controlled.

The success of Color Indexing [14] for recognizing objects from a large database using only multispectral distribution information sparked interest in the development of methods which generalize the basic approach for applicability to environments with changing illumination. Using physical models for multiband image formation, Funt and Finlayson [5] and Healey and Slater [7] developed indexing methods which are relatively insensitive to scene illumination conditions. For the most part, however, these methods disregard image spatial structure and by themselves have been shown to be ineffective for recognition in moderately sized databases [6] [7].

Previous work on illumination-invariant recognition using spatial structure makes use of a multiband correlation model [8]. This model has been exploited successfully for recognition in the presence of large illumination changes. The multiband correlation model, however, represents an image region using six correlation functions which typically contain a large amount of redundant information. A computationally expensive projection method is used for the illumination-invariant comparison of these functions. The complexity of the correlation model transformation in response to illumination changes prohibits the use of this model for the direct recovery of the illumination change matrix.

In this paper, we present a compact multiband image representation based on filtered energy features. We show that matrices of these features transform in the same way as the multiband sensor vectors in response to an illumination change. From this relationship, we derive an efficient pseudoinverse method which simultaneously provides a metric for the illumination-invariant comparison of energy matrices and computes the illumination change matrix. We demonstrate the use of this method for two classes of recognition problems by considering sets of both oriented and rotationally symmetric filters to produce the energy matrices.

2 Modeling Filtered Color Images

Consider a color imaging system that records N measurements at each location (x, y) given by

$$I_i(x,y) = \int_\lambda l(\lambda)s(x,y,\lambda)f_i(\lambda)d\lambda \qquad 1 \leq i \leq N \quad (1)$$

where $s(x,y,\lambda)$ is the spectral reflectance of the surface, $f_i(\lambda)$ is the response of the ith photoreceptor class, and λ denotes wavelength. Denote the multiband image by the vector $I(x,y) = (I_1(x,y), I_2(x,y), \ldots, I_N(x,y))^T$. Let $I(x,y)$ be the image of a surface illuminated by spectral distribution $l(\lambda)$ and let $\tilde{I}(x,y)$ be the image of the same surface illuminated by spectral distribution $\tilde{l}(\lambda)$. Using the linear spectral reflectance model

$$s(x,y,\lambda) = \sum_{1 \leq j \leq N} \sigma_j(x,y)S_j(\lambda) \qquad (2)$$

where the $S_j(\lambda)$ are fixed basis functions, the images are related by a linear transformation [8]

$$I(x,y) = M\tilde{I}(x,y) \qquad (3)$$

where M is an $N \times N$ matrix with elements that depend on $l(\lambda)$ and $\tilde{l}(\lambda)$. The accuracy of the linear model in (2) has been confirmed by several studies [3] [10] [12].

A linear filter $h(x,y)$ can be applied to each band of $I(x,y)$ to obtain a filtered multiband image $O(x,y) = (O_1(x,y), O_2(x,y), \ldots, O_N(x,y))^T$ where

$$O_i(x,y) = I_i(x,y) * h(x,y) \qquad (4)$$

where $*$ denotes two-dimensional convolution. From (3), the filtered images $O(x,y)$ and $\tilde{O}(x,y)$ derived from $I(x,y)$ and $\tilde{I}(x,y)$ are related by the linear transformation M according to

$$O(x,y) = M\tilde{O}(x,y). \qquad (5)$$

Let $h_1(x,y), h_2(x,y), \ldots, h_n(x,y)$ be a set of real positive linear filters $h_i(x,y) \geq 0, (x,y) \in (-\infty, \infty), i = 1, 2, \ldots, n$. Define the energy in a filtered image band by

$$E_{ij} = \sum_{x,y} I_i(x,y) * h_j(x,y) \qquad (6)$$

The energy matrix of the N band image $I(x,y)$ for the set of n filters is given by

$$E = \begin{bmatrix} E_{11} & E_{12} & \cdots & E_{1n} \\ E_{21} & E_{22} & \cdots & E_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ E_{N1} & E_{N2} & \cdots & E_{Nn} \end{bmatrix} \qquad (7)$$

If this set of filters is applied to the multiband images $I(x,y)$ and $\tilde{I}(x,y)$, then from (5) and (6) the corresponding energy matrices are related by

$$E = M\tilde{E} \qquad (8)$$

where M is the linear transformation in (3). This relationship will be used to derive methods for illumination correction and illumination-invariant recognition.

3 Illumination Correction and Recognition

3.1 Estimating Illumination Changes

Given energy matrices E and $\tilde{E}$ corresponding to the same surface under different illuminants, the illumination change matrix M can be estimated. If $n > N$, then (8) is overdetermined and M can be estimated using the pseudoinverse

$$M_p = E\tilde{E}^T\left(\tilde{E}\tilde{E}^T\right)^{-1} \qquad (9)$$

If we define the residual matrix

$$R = E - M\tilde{E} \qquad (10)$$

then M_p is the matrix M which minimizes the sum of the squares of the elements of R. Since M is the matrix which relates the original images in (3), the matrix M_p can be used to transform the image $\tilde{I}(x,y)$ obtained under illumination $\tilde{l}(\lambda)$ to the corresponding image $M_p\tilde{I}(x,y)$ which would be obtained under illumination $l(\lambda)$.

Given two images, an important question for recognition is whether measured energy matrices E_1 and E_2 correspond to the same surface under different illumination conditions. In this context, the residual matrix formed using the pseudoinverse

$$R = E_1 - \left[E_1E_2^T(E_2E_2^T)^{-1}\right]E_2 \qquad (11)$$

is a measure of how well the set of energy vectors for image 1 (columns of E_1) can be related to the set of energy vectors for image 2 (columns of E_2) by a single linear transformation. In section 4, the squared norm of R

$$||R||^2 = \sum_{1 \leq i \leq N} \sum_{1 \leq j \leq n} [R(i,j)]^2 \qquad (12)$$

will be used for the illumination-invariant comparison of energy matrices. Small values of $||R||^2$ indicate that a linear relationship of the form of (8) holds for E_1 and E_2 which is a necessary condition for these energy matrices to correspond to the same surface.

3.2 Filter Selection

The set of filters $h_1(x,y), h_2(x,y), \ldots, h_n(x,y)$ can be selected to extract a wide range of spatial characteristics of an input image. For many applications, the outputs of oriented filters are useful for representing image spatial structure. In this paper, we will examine the use of steerable filters [1] [4] [13] which capture information about the response of a filter at any orientation using a small set of basis filters. Many functions are steerable including all polynomials in x and y multiplied by a rotationally symmetric function [4]. Since

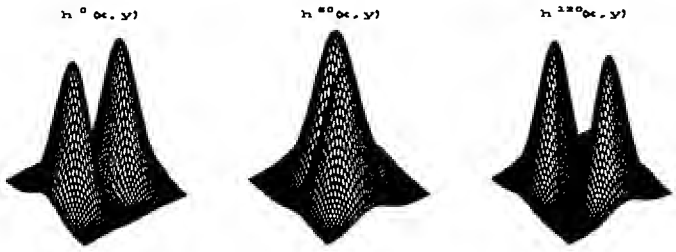


Figure 1: Spatial Response of Basis Filters

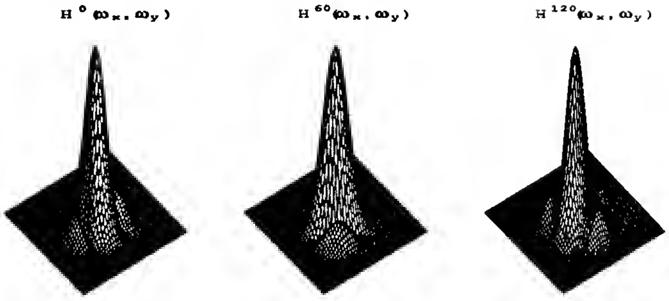


Figure 2: Frequency Response of Basis Filters

the relationship in (8) requires a set of filters with non-negative values, we choose the kernel $x^2 e^{-\frac{x^2+y^2}{2\sigma^2}}$. For this kernel, a filter at an arbitrary orientation θ can be synthesized using a linear combination of three basis filters [4] according to

$$h^\theta(x,y) = k_1(\theta)h^0(x,y)+k_2(\theta)h^{60}(x,y)+k_3(\theta)h^{120}(x,y) \quad (13)$$

where

$$\begin{aligned} h^0(x,y) &= x^2 e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (14)\\ h^{60}(x,y) &= (\frac{1}{2}x + \frac{\sqrt{3}}{2}y)^2 e^{-\frac{x^2+y^2}{2\sigma^2}}\\ h^{120}(x,y) &= (-\frac{1}{2}x + \frac{\sqrt{3}}{2}y)^2 e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

and

$$\begin{aligned} k_1(\theta) &= 1 + 2\cos 2\theta \quad (15)\\ k_2(\theta) &= 1 - \cos 2\theta + \sqrt{3}\sin 2\theta\\ k_3(\theta) &= 1 - \cos 2\theta - \sqrt{3}\sin 2\theta \end{aligned}$$

Thus, the output of these three basis filters determines the output for a filter of arbitrary orientation. Figure 1 shows the basis filters $h^0(x,y), h^{60}(x,y), h^{120}(x,y)$. Figure 2 shows the frequency response of these filters.

The use of oriented filters for recognition requires that the orientation of an object is known. In applications where object orientation is unknown, rotation invariant recognition can be achieved by using filters which are rotationally symmetric. A rotation invariant version of the oriented filter defined by (13) can be constructed by integrating over θ to obtain

$$H(x,y) = \int_\theta h^\theta(x,y)d\theta \quad (16)$$

Figure 3 shows the filter $H(x,y)$ and figure 4 shows its frequency response.

The set of rotationally invariant filters $h_1(x,y), h_2(x,y), \ldots, h_n(x,y)$ which are used for recognition can be constructed using (16) for several values of the scale parameter σ. Following recognition and illumination correction, the rotation angle of an object in the scene can be estimated using oriented filters. The rotation angle estimation process is described in the Appendix.

4 Experimental Results

We conducted two sets of experiments to test the method developed in section 3 for recognition and illumination correction. In the set of experiments described in 4.1, the oriented filters were used for recognition in the presence of illumination changes. In the set of experiments described in 4.2, the rotationally symmetric filters were used for recognition in the presence of combined illumination and rotation changes. In each of these experiments, the squared norm $||R||^2$ defined by (11) and (12) was used for the comparison of energy matrices. To evaluate the effectiveness of the new approach over traditional approaches, we also examined the use of the direct distance between energy matrices E_1 and E_2 defined by

$$||E_1 - E_2||^2 = \sum_{1\le i\le N}\sum_{1\le j\le n}[E_1(i,j) - E_2(i,j)]^2 \quad (17)$$

for recognition.

The multiband images for these experiments were obtained at our image acquisition facility using a Sony XC-77 CCD camera with three Corion filters having the spectral transmission bands: CA-600 (580-700nm), CA-550 (500-600nm), CA-500 (400-520nm) in conjunction with a RasterOps TC-PIP framegrabber. A Newport model 765 tungsten-halogen light source filtered by a Corion BG-38 blue-green filter was used to obtain nearly white illumination. Yellow, red, and green illuminants were obtained by using color transmission filters with the Newport source.

4.1 Illumination Invariant Recognition

A database was formed using images of sixteen objects obtained under white illumination. Each object was represented by the 3×7 energy matrix defined by

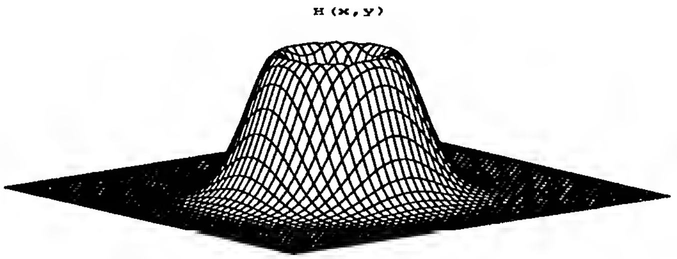


Figure 3: Spatial Response of H(x,y)

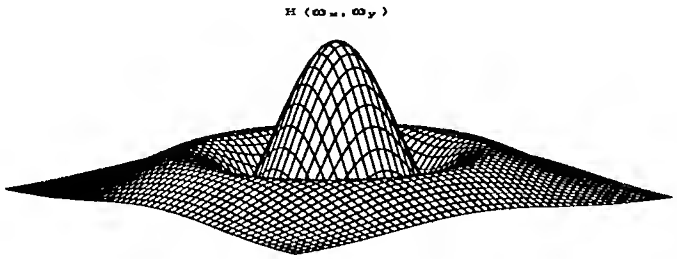


Figure 4: Frequency Response of H(x,y)

(7). Six spatial filters defined by $h^0(x,y), h^{60}(x,y)$, and $h^{120}(x,y)$ of (14) with $\sigma = 2$ and $\sigma = 4$ were used. The filtered energies were augmented by the energies in each of the three unfiltered color bands so that $n = 7$.

A set of forty-eight test images was assembled by imaging each database object under each of yellow, red, and green illumination. Each test image was represented by the 3×7 energy matrix defined above and classified as an instance of the database object with which it has the smallest distance $||R||^2$ defined by (11) and (12). Using this method, each of the 48 test images was classified correctly. For each match, the illumination correction matrix M_p was computed using (9). According to (3), this enables a test image $I_T(x,y)$ to be transformed to its appearance under the database white illumination using $M_p I_T(x,y)$. For comparison, each of the test images was classified using the direct distance $||E_1 - E_2||$ defined by (17). Using this distance, only 5 of the 48 test images were classified correctly. Table 1 shows the distribution of classification ranks for the 48 test images using $||E_1 - E_2||^2$. For example, a rank of 6 for a test image means that the correct match in the database received the sixth smallest value of $||E_1 - E_2||^2$.

Figures 5-10 display several database and test images along with the corresponding illumination corrected images. In each row, the left image is the database image and the right image is the test image. The middle image is the illumination corrected test image. We see that in most cases the illumination corrected test image is visually indistinguishable from the database image.

Table 1: Illumination Invariant Recognition (Direct Distance)

rank	1	2	3	4	5	6	7	8
number	5	4	4	3	5	3	1	3
rank	9	10	11	12	13	14	15	16
number	3	3	3	2	3	2	2	2

The plot in figure 11 shows the set of distances $||R||^2$ computed between a test image of object 2 (peacock) under green illumination and each database object. Large distances for a database object are truncated at the top of the scale. We observe that the distance for the correct match (object 2) is much smaller than the other distances. Such a distribution of distances is typical for this set of experiments.

4.2 Illumination and Rotation Invariant Recognition

The method developed in section 3 can be used for the recognition of objects following illumination changes and plane rotations if a set of rotationally symmetric filters are used. We conducted a set of experiments to study the performance of the algorithm for this case using a database of sixteen objects imaged under white illumination. For this set of experiments, 3×4 energy matrices were used. The matrices were computed using the three rotationally symmetric filters $H(x,y)$ defined by (16) for $\sigma = 2$, $\sigma = 4$, and $\sigma = 8$ in conjunction with the energies in each of the three unfiltered color bands. A set of 64 test images was generated by imaging rotated versions of the database objects under white, yellow, red, and green illumination. As in 4.1, each of the 64 test images was classified as an instance of the database object with which it has the smallest distance $||R||^2$ defined by (11) and (12). Using this approach, each of the 64 test images was classified correctly. For each match of test image to database image, the illumination correction matrix M_p was computed by (9) and the rotation angle was computed by the method described in the Appendix using $h^0(x,y), h^{60}(x,y)$, and $h^{120}(x,y)$ with $\sigma = 8$. Using the direct distance $||E_1 - E_2||^2$ resulted in 22 of the 64 test images receiving the correct classification. Table 2 shows the distribution of classification ranks for the 64 test images using $||E_1 - E_2||^2$. We note that sixteen of the test images are rotated versions of database images under the white database illumination. As might be expected given the use of rotationally symmetric filters, all sixteen of these test images were among the 22 which received the correct classification.

Figures 12-14 display several database images and test images along with the illumination and rotation corrected test images. In each row, the test image is shown on the right and the matching database image is shown on the left. The middle image in each row is the illumination and rotation corrected test image. For each test image, the rotation correction was accurate to within 3°.

5 Summary

We have introduced an energy matrix representation for multiband images defined using a small set of spatial filters. Using an accurate linear model for spectral reflectance, we have shown that energy matrices deform according to a linear transformation in response to a change in the spectral properties of the illumination. This relationship is the basis of a pseudoinverse method for the illumination-invariant comparison of energy matrices which also recovers the illumination change relating the original multiband images. By using rotationally symmetric filters, the method can be used for the recognition of rotated objects in the presence of illumination changes. A large set of experiments demonstrates the effectiveness of the new algorithm compared to traditional energy matching approaches.

Appendix: Rotation Angle Estimation

Consider a pair of color images $I(x,y)$ and $\widetilde{I}(x,y)$ which are related by an illumination change and a rotation. Using a set of rotationally symmetric filters, the illumination change M_p can be estimated using (9) and the illumination corrected image $M_p\widetilde{I}(x,y)$ will be related to $I(x,y)$ by a rotation.

Using the filter $h^\theta(x,y)$ defined by (13), let the orientation θ_i of an image band $I_i(x,y)$ be the value of θ for which the energy

$$E_i(\theta) = \sum_{x,y} I_i(x,y) * h^\theta(x,y) \tag{18}$$

is maximum. From (13), the oriented energy is given by

$$E_i(\theta) = k_1(\theta)E_i(0^\circ) + k_2(\theta)E_i(60^\circ) + k_3(\theta)E_i(120^\circ) \tag{19}$$

which is maximized for

$$\theta_i = \frac{1}{2}\arctan\left(\frac{\sqrt{3}(E_i(60^\circ) - E_i(120^\circ))}{2E_i(0^\circ) - E_i(60^\circ) - E_i(120^\circ)}\right) \tag{20}$$

Thus, the orientation is determined using the output of only three filters.

For N bands the orientation vector is given by $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$. Given orientation vectors θ and $\widetilde{\theta}$ corresponding to images $I(x,y)$ and $M_p\widetilde{I}(x,y)$, the rotation angle α is estimated as the average relative rotation over the N bands using

$$\alpha = \frac{1}{N}\sum_{1\leq i\leq N}(\theta_i - \widetilde{\theta}_i) \tag{21}$$

Acknowledgment

This work has been supported in part by the Office of Naval Research under grant N00014-93-1-0540.

References

[1] W. Beil. Steerable filters and invariance theory. *Pattern Recognition Letters*, 15:453–460, May 1994.

[2] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(1):55–73, 1990.

[3] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychonomic Sci.*, 1:369, 1964.

[4] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(9):891–906, September 1991.

[5] B. Funt and G. Finlayson. Color constant color indexing. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(5):522–528, May 1995.

[6] G. Healey and A. Jain. Retrieving multispectral satellite images using physics-based invariant representations. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8):842–848, August 1996.

[7] G. Healey and D. Slater. Global color constancy: recognition of objects by use of illumination invariant properties of color distributions. *J. Opt. Soc. Am. A*, 11(11):3003–3010, November 1994.

[8] G. Healey and L. Wang. Illumination-invariant recognition of texture in color images. *J.Opt. Soc. Am. A*, 12(9):1877–1883, September 1995.

[9] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24:1167–1186, 1991.

[10] L. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10):1673–1683, October 1986.

[11] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8):837–842, August 1996.

[12] J.P.S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of the Optical Society of America A*, 6:318–322, 1989.

[13] P.Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(5):488–499, 1995.

[14] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

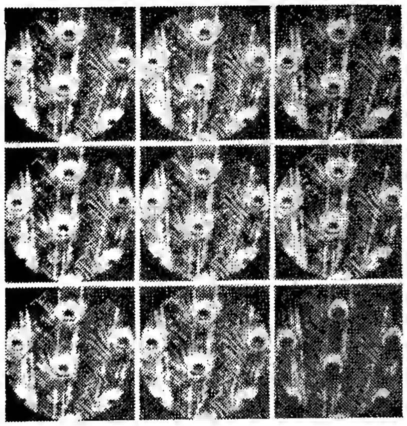


Figure 5: Peacock

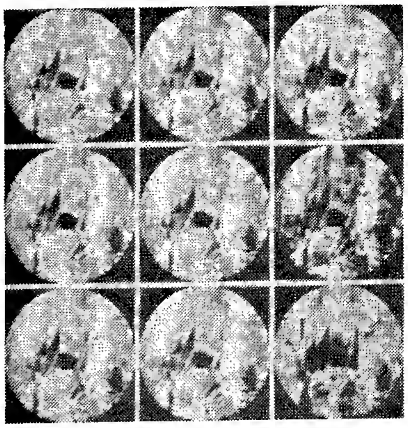


Figure 8: Painted Pattern

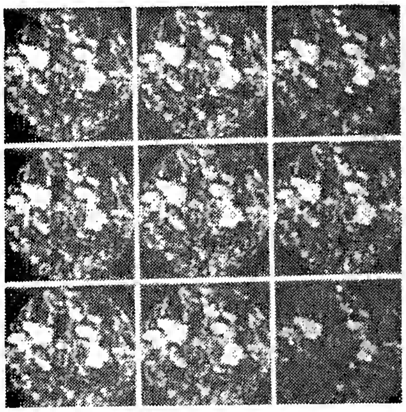


Figure 6: Flowers

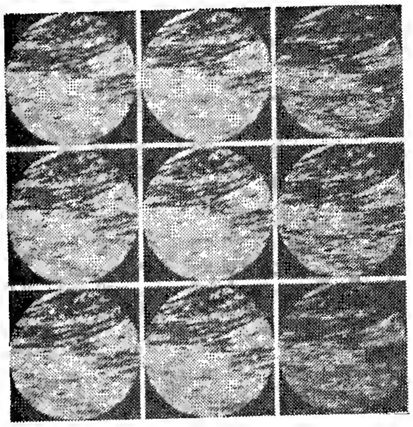


Figure 9: Box

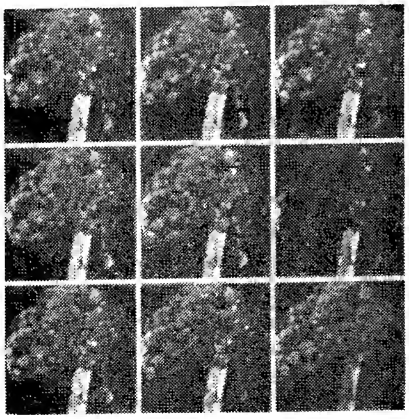


Figure 7: Trees

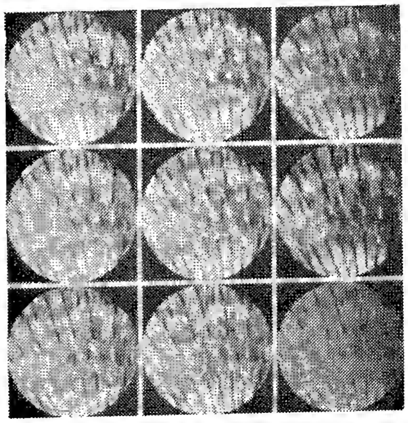


Figure 10: Shell

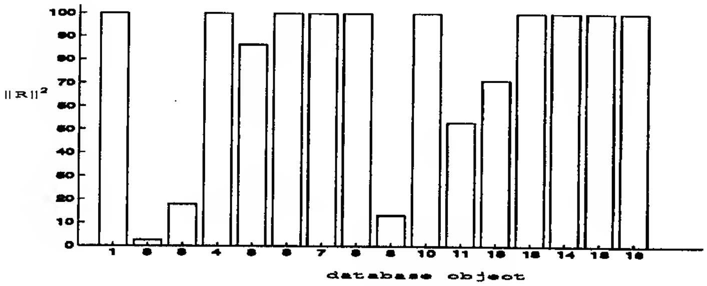


Figure 11: Database distances for object2 under green illumination

Table 2: Illumination/Rotation Invariant Recognition (Direct Distance)

rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
number	22	6	2	4	3	3	6	2	1	1	1	3	3	3	2	2

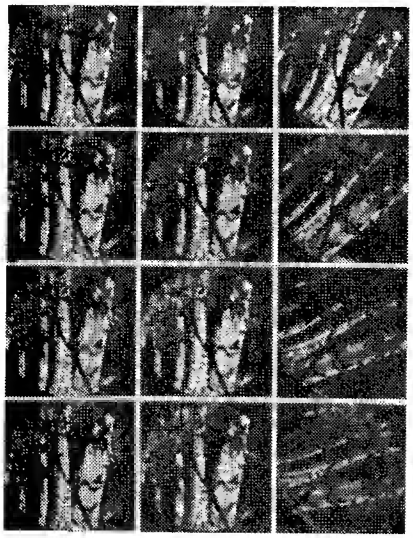


Figure 12: Trees

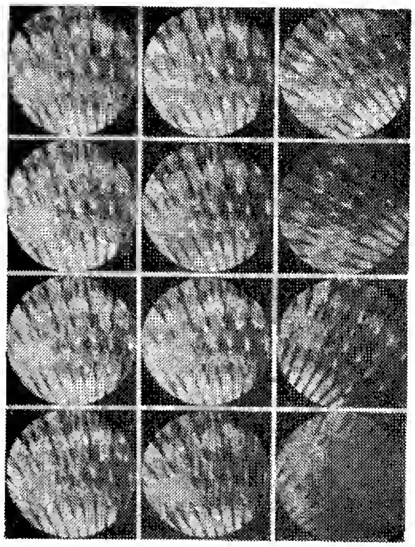


Figure 13: Shell

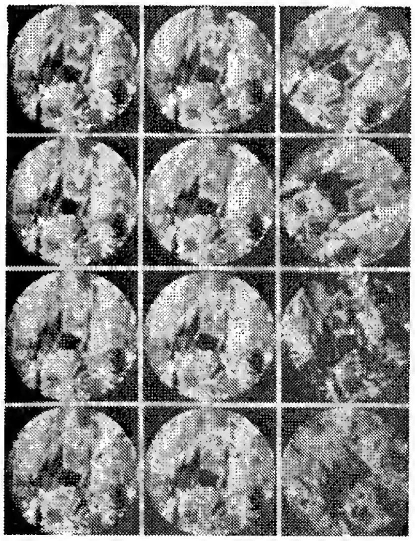


Figure 14: Painted Pattern

Using spectral/spatial information for automatic recognition

Glenn Healey and David Slater
Computer Vision Laboratory
Electrical and Computer Engineering
University of California
Irvine, CA 92697
healey,dslater@ece.uci.edu
http://www.cvl.uci.edu

Spatial filters provide a useful and efficient means of analyzing an input color image into components which capture different spatial properties. Representations based on spatial filtering have restricted usefulness for recognition, however, because the output of a spatial filter across an image depends on the scene illumination conditions. In this paper, we use a physically accurate linear model for spectral reflectance to derive invariants of distributions in spatially filtered color images that do not depend on the scene illumination. These invariants can be used for the illumination-invariant recognition of regions following an arbitrary linear filtering operation. We describe a method for illumination correction based on color distributions and introduce an illumination change consistency constraint which is useful for verifying matches obtained using the invariants. We show using a set of classification experiments that the filtered distribution invariants can significantly improve the capability of a recognition system in environments where illumination cannot be controlled.

1 Introduction

The measured color distribution in the image of an object provides a useful source of information for recognition. Swain and Ballard [13] have shown, for example, that color distributions can often be used directly for recognition without using corresponding geometric information. Such a method is effective in situations where the illumination spectral content is held constant. As the illumination environment changes, however, the color distribution measured in the image changes limiting the usefulness of this approach.

Recent work on color constancy has focused on computing illumination-invariant descriptors of color image regions. Funt and Finlayson [3] introduced a method called Color Constant Color Indexing that matches distributions of color ratios which are illumination-invariant under the coefficient model of sensor response. Healey and Slater [6] used a linear spectral reflectance model to derive a set of moment invariants of color distributions that does not depend on the illumination. Each of these methods has been shown to improve recognition accuracy significantly in the presence of illumination changes over direct distribution matching. The representations used by both methods, however, fail to capture most of the spatial information in a color image. This is perhaps an important reason why these methods have been shown to lack the discriminatory power necessary for recognition in large databases [5] [6].

Image models which characterize spatial structure provide important information that is not captured by color distributions. Spatial structure is particularly significant for textured regions which often occur in images of natural scenes. The analysis of gray-scale texture has received significant attention over many years, e.g.[1] [4], whereas considerably less effort has been devoted to the study of spatial structure in color images. Recently, the effectiveness of a color correlation model for illumination-invariant recognition has been demonstrated on a set of color textures [7]. The disadvantage of this approach is that the representation includes six correlation functions and recognition is achieved by the projection of these functions onto the database correlation representation. For recognition in large databases, efficiency considerations suggest the use of a more compact color image representation.

Spatial filters can be used to capture spatial properties of interest in a color image. Filter-based image representations have the advantage that they can be computed efficiently and component filters can be selected to optimize performance for specific applications. In the context of recognition, however, the outputs of spatial filters depend on the illumination conditions. In this paper, we show that color distributions in the filtered image of a surface undergo a linear coordinate transform in response to illumination changes. This relationship is used to derive illumination invariants of distributions in spatially filtered color images. This provides a powerful mechanism for recognizing a wide range of spatial patterns in color images under unknown illumination conditions. In addition, we show that combining descriptors over several filtered distributions leads to significantly better recognition rates than can be achieved by using a single filtered distribution.

2 Representing Color Images

At each location (x, y) a color imaging system obtains the n measurements

$$\rho_i(x,y) = \int_\lambda l(\lambda) s(x,y,\lambda) f_i(\lambda) d\lambda \qquad 1 \leq i \leq n \quad (1)$$

where $l(\lambda)$ is the spectral power distribution of the scene illumination, $s(x, y, \lambda)$ is the spectral reflectance of the surface, $f_i(\lambda)$ is the sensitivity of the ith sensor class, and λ denotes wavelength.

The spectral reflectance at each location (x, y) can be approximated by

$$s(x,y,\lambda) = \sum_{1 \leq j \leq m} \sigma_j(x,y) S_j(\lambda) \qquad (2)$$

where the $S_j(\lambda)$ are a set of m fixed basis functions. Such approximations have been used previously in color vision and several researchers have analyzed their accuracy for large sets of spectral reflectance functions [2] [8] [9]. Three properly chosen basis functions have been shown to provide accurate approximations to a large range of spectral reflectance functions.

We consider the case where $m = n$. Let $\rho(x, y) = (\rho_1(x, y), \rho_2(x, y), \ldots, \rho_n(x, y))^T$ denote the column vector of sensor measurements and let $\sigma(x, y) = (\sigma_1(x, y), \sigma_2(x, y), \ldots \sigma_n(x, y))^T$ denote the column vector of spectral reflectance function weights. Then

$$\rho(x,y) = A\sigma(x,y) \qquad (3)$$

where A is an $n \times n$ matrix with elements

$$A_{kj} = \int l(\lambda) S_j(\lambda) f_k(\lambda) d\lambda \qquad (4)$$

Thus, $\sigma(x, y)$ describes the surface spectral reflectance across the image and A depends on the illumination $l(\lambda)$ but not on image location.

Suppose that two color images are taken of a scene. The first image $\rho(x, y)$ is obtained using illumination with spectral distribution $l(\lambda)$ and the second image $\tilde{\rho}(x, y)$ is obtained using illumination with spectral distribution $\tilde{l}(\lambda)$. The two images are described by

$$\rho(x,y) = A\sigma(x,y), \qquad \tilde{\rho}(x,y) = \tilde{A}\sigma(x,y). \qquad (5)$$

If the matrices A and $\tilde{A}$ corresponding to $l(\lambda)$ and $\tilde{l}(\lambda)$ are nonsingular, then

$$\tilde{\rho}(x,y) = M\rho(x,y) \qquad (6)$$

where M is an $n \times n$ matrix.

Suppose we apply the linear filter $g(x, y)$ to each band of $\rho(x, y)$ and $\tilde{\rho}(x, y)$ to obtain

$$\rho'(x,y) = (\rho_1(x,y) * g(x,y), \ldots, \rho_n(x,y) * g(x,y))^T \qquad (7)$$

$$\tilde{\rho}'(x,y) = (\tilde{\rho}_1(x,y) * g(x,y), \ldots, \tilde{\rho}_n(x,y) * g(x,y))^T \qquad (8)$$

where $*$ denotes convolution. From (7) and (8), we can write

$$\tilde{\rho}'(x,y) = M\rho'(x,y) \qquad (9)$$

Let $H'(\cdot)$ be the histogram having an n-dimensional argument describing the color distribution in the filtered color image $\rho'(x, y)$. Let $\tilde{H}'(\cdot)$ be the histogram having an n-dimensional argument describing the color distribution in the filtered color image $\tilde{\rho}'(x, y)$. Then from (9)

$$\tilde{H}'(M\rho) = H'(\rho) \qquad (10)$$

Thus, an illumination change causes distributions in the filtered images to deform according to a linear coordinate transformation. This relationship will be used for the derivation of illumination invariants in the next section.

3 Illumination Invariants

3.1 Distribution Invariants

We define illumination invariants as numbers which are computed from an image region which do not depend on the illumination $l(\lambda)$. Let $H(\cdot)$ be the histogram having an n-dimensional argument describing the color distribution in a region of the image $\rho(x, y)$. Let $\tilde{H}(\cdot)$ be the histogram having an n-dimensional argument corresponding to the same region of the image $\tilde{\rho}(x, y)$. Then from (6), the measured color pixel distribution for the region will deform according to a linear coordinate transformation in response to an illumination change as given by

$$\tilde{H}(M\rho) = H(\rho) \qquad (11)$$

Using this relationship, we have developed an efficient method for computing vectors of illumination invariants from color distributions [6]. The method is based on a technique for computing affine algebraic moment invariants of functions proposed for shape recognition by Taubin and Cooper [14]. The illumination-invariant vectors can be computed efficiently in time proportional to the number of pixels that define an image region under consideration.

The method for computing moment invariants of a distribution $H(\rho)$ begins by transforming $H(\rho)$ to a normalized distribution $H(L\rho)$ whose second order centered moment matrix is the identity matrix. A Cholesky decomposition is employed to compute L from the second order centered moment matrix of $H(\rho)$. If such a normalization is applied to distributions which were originally related by a linear coordinate transformation, then the normalized distributions will be related by an orthogonal coordinate transformation. Eigenvalues of a moment matrix of the normalized distribution will be invariant to the original linear coordinate transformation and equivalently the illumination change. Distribution invariants have been used as a feature for image database annotation [10] and for the illumination-invariant content-based retrieval of regions in multispectral satellite images [5]. A full description of the computation of these invariants is given in [6].

3.2 Filtered Distribution Invariants

Although distribution invariants have been demonstrated as useful features for recognition, color distributions do not preserve information about the spatial structure in a color image. Thus, distribution invariants cannot discriminate regions with significantly different spatial properties if the regions have similar color distributions. In experiments with a large database of multispectral satellite images [5], for example, distribution invariants were shown to be useful as a filter to reduce the set of candidate matches, but by themselves lacked the discriminatory power for recognition in many cases.

Color images can be analyzed efficiently into components corresponding to different spatial frequency characteristics using a bank of spatial filters. This approach has been applied successfully by many systems for the analysis of gray-scale images. The filter-based representation has the advantage that filters can be selected to emphasize properties of interest which optimize performance for particular applications. In recognition systems, filter-based representations can provide effective features for database indexing. Unfortunately, as shown in section 2, distributions in spatially filtered images depend on the scene illumination.

From (10), we see that filtered color image distributions are related by a linear coordinate transformation. Thus, the moment invariants described in 3.1 will provide illumination-invariant descriptors of distributions in filtered color images. These filtered distribution invariants capture information about spatial structure in an image according to the spatial response of the filter used. Therefore, filtered distribution invariants can be combined with distribution invariants for illumination-invariant recognition. While color distributions are invariant to rotation, scale, and orientation in a scene [12], geometric invariance does not necessarily hold for filtered distributions. A change of scale, for example, alters the spatial frequencies which are observed in an image and a rotation changes the orientation of spatial features. Rotation invariance can be achieved, however, by selecting filters which are rotationally invariant.

We illustrate the use of filtered distribution invariants in figures 1-7. Figure 1 is a color image of a photograph of a tiger under each of four illumination conditions. Moving clockwise from the upper left are images obtained under white, red, yellow, and green illumination. Figure 2 displays each of these images following a lowpass Gaussian filtering on each band. Figure 3 displays each of these images following a highpass difference of Gaussians filtering on each band. Figures 4 - 6 are corresponding images of bushes. Figure 7(a) is a plot of the distribution invariants computed for each of the four unfiltered tiger and bush images. Six invariants are computed for each image but only the three largest are used for display purposes. Figure 7(b) plots the filtered distribution invariants for the lowpass images and figure 7(c) plots the filtered distribution invariants for the highpass images. In each case the invariants for the different objects form reasonably compact separated clusters in the invariant spaces. Since each invariant space captures different information, the use of different filters increases the amount of discriminatory power that is available for recognition.

In figures 7-9, we demonstrate the use of filtered distribution invariants for discriminating patterns with similar color distributions. Figure 8 is a single color pattern under the four different illumination conditions (white, red, yellow, green). Figure 9 is a similar color pattern also imaged under each of the four different illumination conditions. The color distributions for the two patterns are nearly identical and consequently the distribution invariants are unable to discriminate the patterns. This is shown by the overlap of the 'o' (pattern 1 invariants for each illumination condition) and '+' (pattern 2 invariants for each illumination condition) symbols for the unfiltered distribution invariants in figure 7(d). However, invariants computed from a highpass filtered version of the image regions in figures 8 and 9 separate well as shown in figure 7(d). Therefore, filtered distribution invariants can be used to recognize color patterns in the presence of illumination changes based on subtle spatial differences even when the structure of the multispectral distributions does not allow recognition.

4 Illumination Correction

From (6), a scene illumination change transforms measured color pixel vectors according to multiplication by a matrix M. Given a database color pixel distribution and an observed color pixel distribution, the matrix M relating the two distributions can be computed. This illumination correction matrix can be used for several purposes. In [12], M was used during recognition to correct the illumination change relating a candidate matching region in an image and a database region. This illumination correction step allowed a direct comparison of candidate matching regions during hypothesis verification. The illumination correction matrix can also be updated continuously using color changes for a known object in the scene to correct images for scene illumination changes. In 4.1 we describe a method for computing the illumination correction matrix M. In 4.2 we introduce an illumination change consistency constraint which can be used to verify a match hypothesized by a combination of distribution and filtered distribution invariants.

4.1 Computing Illumination Changes

The method for computing the illumination change matrix M proceeds as follows. Let $H(\rho)$ denote the color pixel distribution of a database region and let $\widetilde{H}(\rho)$ denote the color pixel distribution of an observed image region. If the two regions correspond to the same surface area under different illuminants, then $\widetilde{H}(M\rho) = H(\rho)$ as in (11). Transforming the original distributions by the matrices L and $\widetilde{L}$ described in 3.1 will generate normalized distributions $G(\rho) = H(L\rho)$

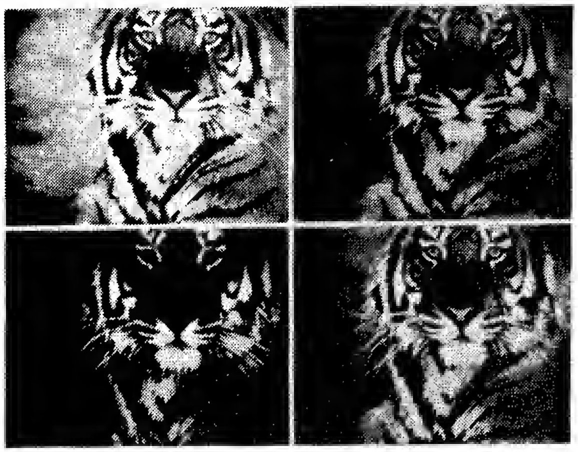


Figure 1: Tiger under four illumination conditions

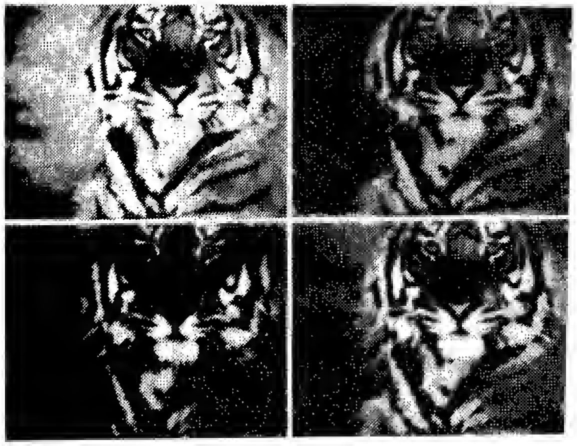


Figure 2: Lowpass filtered tiger images

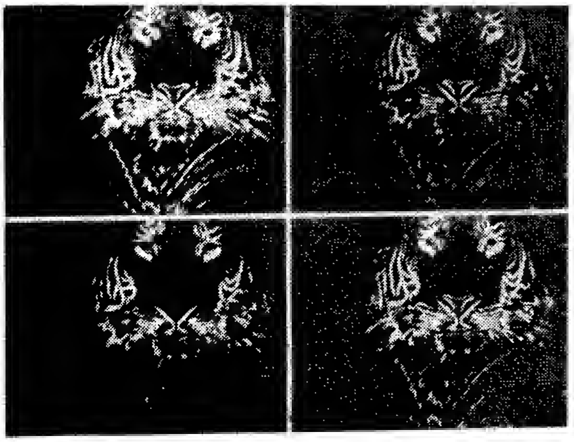


Figure 3: Highpass filtered tiger images

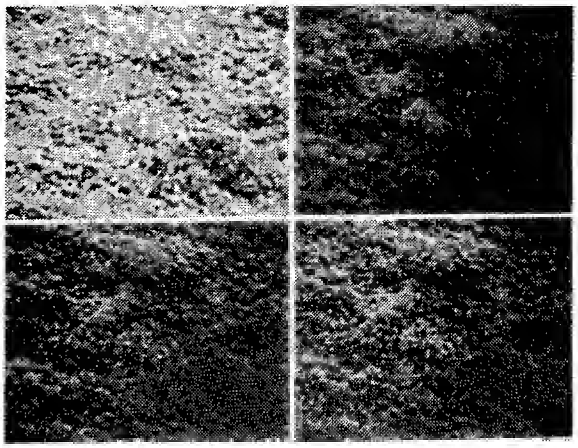


Figure 4: Bushes under four illumination conditions

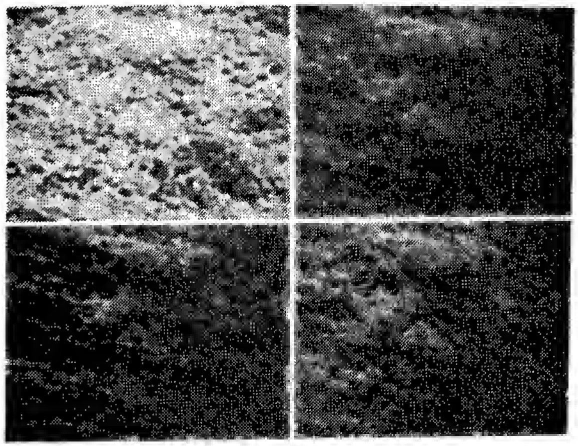


Figure 5: Lowpass filtered bush images

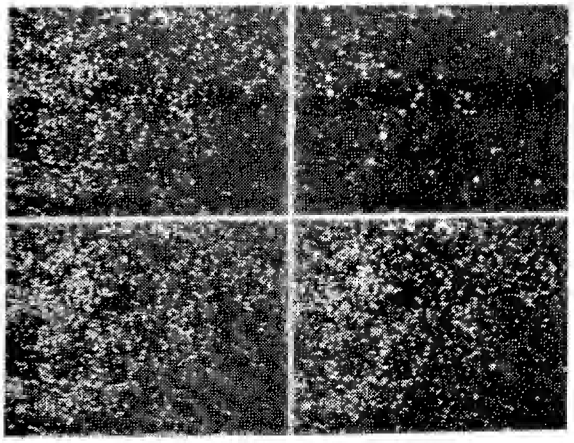


Figure 6: Highpass filtered bush images

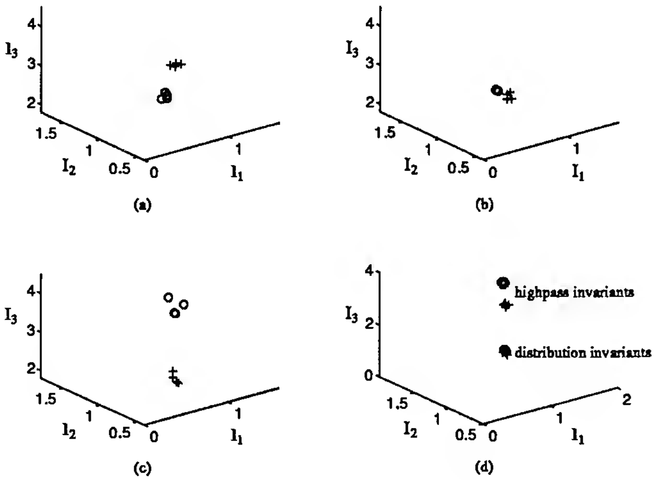


Figure 7: (a) Invariants for unfiltered images 'o' - tiger, '+' - bushes, (b) Invariants for lowpass images 'o' - tiger, '+' - bushes, (c) Invariants for highpass images 'o' - tiger, '+' - bushes, (d) Computed Invariants 'o' - pattern 1, '+' - pattern 2

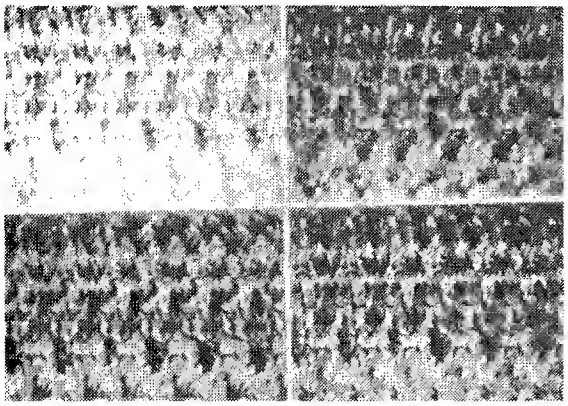


Figure 8: Pattern 1 under four illumination conditions

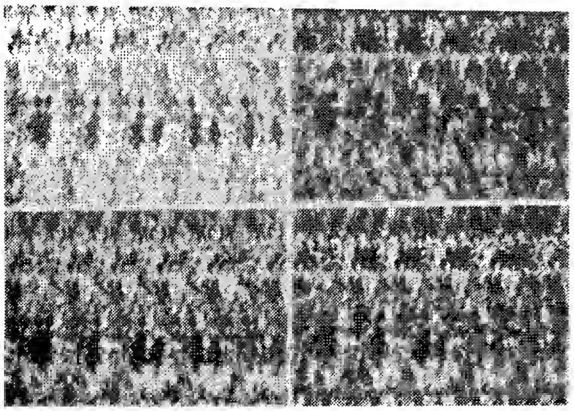


Figure 9: Pattern 2 under four illumination conditions

and $\widetilde{G}(\rho) = \widetilde{H}(\widetilde{L}\rho)$ which are related by an orthogonal coordinate transformation

$$\widetilde{G}(\rho) = G(O\rho) \tag{12}$$

The matrix O is determined by finding an intrinsic coordinate system for the distributions $G(\rho)$ and $\widetilde{G}(\rho)$ and then computing the rotation which aligns these systems. The simplest intrinsic coordinate system is defined by the eigenvectors of a distribution's second order moment matrix. This coordinate system cannot be used for alignment in this case, however, because the second order moment matrices of $G(\rho)$ and $\widetilde{G}(\rho)$ are identity matrices following the normalization by L and $\widetilde{L}$ [14]. The intrinsic coordinate system for each distribution is defined instead using another symmetric 3×3 matrix B which is formed from distribution moments and defined in [11].

The matrices B and $\widetilde{B}$ are computed from $G(\rho)$ and $\widetilde{G}(\rho)$ respectively. These matrices determine intrinsic coordinate systems for the respective distributions and can be aligned to determine the matrix O. Let $\alpha_1, \alpha_2, \alpha_3$ be orthonormal eigenvectors of B sorted by eigenvalue magnitude and let $\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3$ be similarly sorted orthonormal eigenvectors of $\widetilde{B}$. Let C be the 3×3 matrix whose ith row is given by α_i and let $\widetilde{C}$ be the corresponding matrix of $\widetilde{\alpha}_i$'s. Then

$$O = C\widetilde{C}^{-1} \tag{13}$$

is the matrix which relates $G(\rho)$ and $\widetilde{G}(\rho)$ and

$$M = \widetilde{L}\widetilde{C}C^{-1}L^{-1} \tag{14}$$

is the illumination change matrix which relates $H(\rho)$ and $\widetilde{H}(\rho)$.

4.2 Illumination Change Consistency

As described in section 3, a match of distribution invariants combined with matches of one or more sets of filtered distribution invariants provides strong evidence for illumination-invariant recognition. In this subsection, we introduce an illumination change consistency constraint which can be used to further verify a hypothesized match based on more than one vector of invariants. Consider a surface viewed under two different illumination conditions. From (10) and (11), we see that the illumination change matrix relating the color distributions and any filtered distributions is the same matrix M. Therefore, for a hypothesized match based on distribution invariants and filtered distribution invariants, the matrices M relating each pair of corresponding distributions will be the same. This constraint can be applied by evaluating the similarity among the set of illumination correction matrices $M_1, M_2, \ldots, M_N$ which are computed using the method in 4.1 for each of the N distributions associated with an image region. In this work, we consider the distance defined by

$$D = \sum_{k,l} \max_{i,j}\{(m_i(k,l) - m_j(k,l))^2\} \tag{15}$$

where $m_i(k,l)$ is the element at row k, column l of matrix M_i. Thus, D measures the similarity among a set of matrices by evaluating the sum of the maximum square differences between corresponding matrix elements. The illumination change consistency constraint will be examined experimentally in the next section.

5 Experimental Results

We tested the filtered distribution invariants on a database of sixteen objects imaged under different illumination conditions. The images were taken from existing distribution and texture databases [6] [7]. The database consisted of an image of each object taken under nearly white illumination generated by a Newport model 765 tungsten-halogen light source filtered by a Corion BG-38 blue-green filter. A set of forty-eight test images was obtained by imaging each database object under each of yellow, red, and green illumination. The images were captured at our image acquisition facility using a monochrome Sony XC-77 CCD camera in conjunction with three Corion filters having the spectral sensitivity bands: CA-600 (580nm-700nm), CA-550 (500nm-600nm), CA-500 (400nm-520nm) and a RasterOps TC-PIP framegrabber. The imaging system was configured for a linear response which has been verified using patches of known reflectance and neutral filters of known transmission. The images shown in figures 1, 4, 8, and 9 were included in the experiments. Figures 10 and 11 show two of the other images. In each case, the database image obtained under white illumination is shown in the upper left. The test image under red illumination is shown in the upper right, yellow illumination in the lower right, and green illumination in the lower left. Each of the underlying objects is a photograph which was imaged under the four different illuminants.

The invariants used for representing each image in the experiments were computed from 1) the image color distribution $H(\rho)$, 2) the color distribution in a lowpass filtered image $H_l'(\rho)$, and 3) the color distribution in a highpass filtered image $H_h'(\rho)$. A 2-D Gaussian with $\sigma = 1.5$ pixels was used for the lowpass filter and a 2-D difference of Gaussians with $\sigma_1 = 1.5$ pixels and $\sigma_2 = 0.75$ pixels was used for the highpass filter. Each filter was represented using a 9×9 mask. A vector of six invariants was computed for each of the original and filtered images yielding a total of eighteen distribution and filtered distribution invariants for each image. Recognition experiments were conducted in which each of the forty-eight test images was classified as an instance of the database image with which it had the smallest Euclidean distance in invariant space. First, we considered each invariant space separately so that a 6-dimensional vector was used for indexing. The correct classification rates using each of the individual invariant spaces were: unfiltered space (40/48), lowpass space (35/48), highpass space (35/48). Second, we combined all three invariant vectors for each image. The invariant vectors were compared using the Euclidean distance between 18-dimensional concatenated vectors. Using the combination of the three invariant vectors for each object yielded a correct classification rate of 47/48. The only test image which was classified incorrectly was texture 1 under red illumination (figure 10) which was classified as an instance of texture 2 (figure 11). These textures have similar color and spatial properties. We note that recognition accuracy is significantly improved by combining invariants corresponding to color distribution and different filtered distributions.

We examined the use of the illumination change consistency constraint derived in 4.2 for verifying matches when the vector of 18 invariants generated more than one close match. All invariant distances between a test image and a database image which were within 0.10 of the best match were used to signify a candidate match. Using this criteria, six of the forty-eight test images had more than one candidate match in the database. One of these six test images was the single misclassified image using the combined invariant vector distance. For each of these images, the distance D of (15) was computed for each candidate match. The candidate match with the smallest D was classified as the database match. Using this procedure, each of the six test images with more than one candidate match was classified correctly. Therefore, using the invariant distance to generate candidate matches and the illumination change consistency constraint for verification lead to an overall correct classification rate of 48/48.

Using this set of images, we also studied the behavior of the illumination invariants with respect to the spatial filtering operations. For this purpose, we define the dispersion d of a set of invariant vectors $V_1, V_2, \ldots, V_P$ by

$$d = \frac{1}{P} \sum_{1 \le i \le P} ||V_i - \overline{V}||^2 \qquad (16)$$

where $\overline{V}$ is the mean vector of $V_1, V_2, \ldots, V_P$. We computed d for each set of four distribution invariant vectors corresponding to a single object under the different illumination conditions. Thus, for perfect illumination invariance d would be zero for each object. We also computed d for each set of four invariant vectors corresponding to the lowpass filtered versions of a single object under the different illumination conditions. Similarly, we computed d for each set of four invariant vectors corresponding to the highpass filtered versions of each object. The average dispersion values over the 16 objects were: raw images $d_r = 0.322$, lowpass images $d_l = 0.377$, highpass images $d_h = 0.647$. The larger dispersion value for the highpass images can be attributed to the fact that a significant fraction of the pixels in the highpass images have values near zero which have little effect on the distribution moments. Thus, for the highpass images smaller distribution samples contribute to the moment computation leading to greater dispersion of the invariants.

6 Discussion

The spatial structure invariants introduced in this paper have many applications. These invariants allow for the efficient illumination-invariant comparison

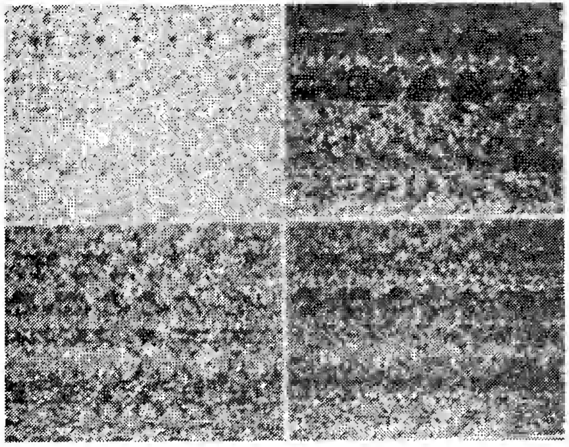


Figure 10: Texture 1

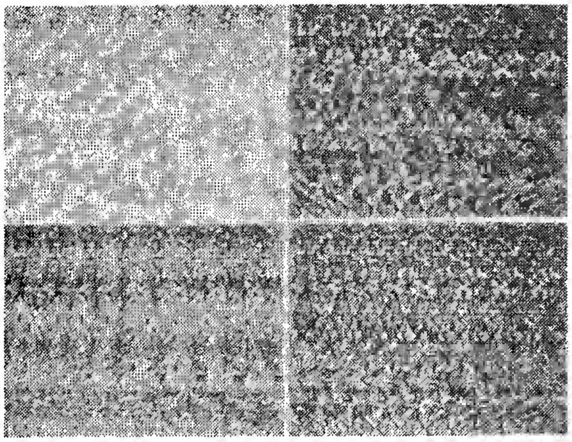


Figure 11: Texture 2

of regions following an arbitrary linear filtering operation. Therefore, these invariants can be used for recognizing a wide range of deterministic and random patterns in color images. As demonstrated in section 5, incorporating spatial information by combining invariants of filtered color distributions $H'(\rho)$ with invariants of the image color distribution $H(\rho)$ can significantly improve the performance of recognition. An important issue for future work is the selection of the set of spatial filters $g(x, y)$ that is used to generate the filtered color images $\rho'(x, y)$ that are used to compute invariants. Filters can be designed that extract a wide variety of spatial properties of an input image. General considerations suggest the use of a set of filters that provides a compact representation that is descriptive enough to enable accurate recognition. For particular recognition problems, optimized filter sets can be designed that maximize object discriminability while using a total number of invariants that is as small as possible.

Acknowledgement

This work has been supported in part by the Office of Naval Research under grant N00014-93-1-0540.

References

[1] R. Chellappa and A.K. Jain, editors. *Markov Random Fields, theory and applications.* Academic Press, San Diego, 1993.

[2] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychonomic Sci.*, 1:369, 1964.

[3] B. Funt and G. Finlayson. Color constant color indexing. *IEEE Trans. Patt. Anal. Machine Intell.*, 17:522–529, 1995.

[4] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5), May 1979.

[5] G. Healey and A. Jain. Retrieving multispectral satellite images using physics-based invariant representations. *IEEE Trans. Patt. Anal. Machine Intell.*, 18(8):842–848, August 1996.

[6] G. Healey and D. Slater. Global color constancy: recognition of objects by use of illumination-invariant properties of color distributions. *J. Opt. Soc. Am. A*, 11(11):3003–3010, November 1994.

[7] G. Healey and L. Wang. Illumination-invariant recognition of texture in color images. *J.Opt. Soc. Am. A*, 12(9):1877–1883, September 1995.

[8] L. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *J. Opt. Soc. Am. A*, 3(10):1673–1683, October 1986.

[9] J.P.S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *J. Opt. Soc. Am. A*, 6:318–322, 1989.

[10] R.W. Picard and T.P. Minka. Vision texture for annotation. *Multimedia systems*, 3:3–14, 1995.

[11] D. Slater and G. Healey. Combining color and geometric information for the illumination invariant recognition of 3-D objects. Technical Report ECE-94-12-01, University of California, Irvine, 1994.

[12] D. Slater and G. Healey. The illumination-invariant recognition of 3D objects using local color invariants. *IEEE Trans. Patt. Anal. Machine Intell.*, 18(2):206–210, February 1996.

[13] M. Swain and D. Ballard. Color indexing. *Int. J. Comp. Vision*, 7:11–32, 1991.

[14] G. Taubin and D. Cooper. Object recognition based on moment (or algebraic) invariants. In J. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 375–397. MIT Press, Cambridge, Mass, 1992.

A Neural Network Approach to Indexing *

Mark R. Stevens J. Ross Beveridge Charles W. Anderson
Colorado State University
stevensm/ross/anderson@cs.colostate.edu
http://www.cs.colostate.edu/~vision

Abstract

Template matching is an effective but computationally expensive means of locating vehicles in imagery. To reduce processing time, we use neural networks to predict a small subset of templates most likely to match each image chip. Results on actual LADAR images show that limiting the templates to those selected by the network reduces the computation time by a factor of 5 without sacrificing identification accuracy.

1 Introduction

Neural networks are often used to extract complex, nonlinear relationships from a data set. In this paper, we use this nonlinear mapping to reduce the computation associated with template matching. Template matching requires the application of numerous templates to rectangular image chips [Perlovsky *et al.*, 1995; Katz *et al.*, 1993]. Each template corresponds to a particular type of vehicle at a specific orientation. After all templates are applied, the best matched templates are returned as the most likely vehicles present [Bevington, 1992].

In order to detect a wide range of vehicle types, a large number of templates must be applied. For this method to be feasible, computation must be fairly efficient. Unfortunately, computation time is usually $O(n)$ where n is the number of templates. Two methods are typically used for reducing processing time: parallel hardware is added [Fang *et al.*, 1987], or focus-of-attention mechanisms reduce the number of image chips [Beveridge *et al.*, 1994a].

A different approach is to index a subset of the possible templates to apply. Here we describe how neural networks can be used to predict the utility of applying each template to a given image chip. Then only those templates expected to match well need to be applied. We show that our neural network indexing reduces the computation time by a factor of 5 without sacrificing identification accuracy on a set of real images. A more detailed explanation can be found in [Stevens *et al.*, 1997].

2 Neural Networks and Template Probing

Our neural network indexing method is based on template probing. Template probing is a correlation technique which matches stored templates to image chips. Each template represents the vehicle at a specific location, and consists of a set of probe test points for locating depth discontinuities along the object contour. These templates are derived off-line from rendered images of 3D CAD models at specific locations and orientations in the image. From these renderings a co-occurrence matrix is derived.

A co-occurrence matrix is a sparse, binary ma-

* This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office.

trix of size $n \times n$ where n is the number of pixels in the rendered image. Matrix elements are 1 where the probe test points have a difference in depth greater than some threshold. When a new image chip is input, a co-occurrence matrix is formed and exhaustively compared to all the templates. The ratio of similarities to total comparisons gives a quality of match score.

Instead of the exhaustive comparison, we use a neural network to predict the 25 best templates (out of 3, 800) to apply to each chip. Processing time is greatly reduced at the expense of adding the small amount of internal network calculations. Each network (one per vehicle) receives the sparse co-occurrence matrix as input. The outputs are predictions of the degree of match between the matrix and each template. Rather than requiring the networks to learn accurate predictions, we only assume they are correct in relative ranking.

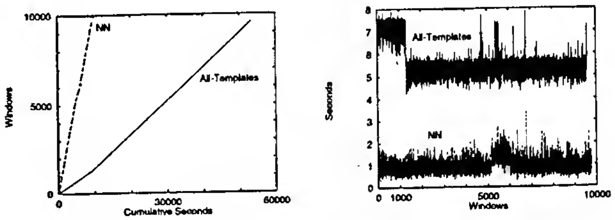
Each network is trained with standard error back propagation [Rumelhart *et al.*, 1986]. The training data was generated from 250 synthetic renderings of the various (3 were used here) vehicles placed in the scene at random orientations. For each chip in these images, the co-occurrence matrix is used with the exhaustive technique to generate the match scores. The training set is formed by pairing each matrix with the match scores. The best network topology and learning parameters were found empirically by training the system and observing performance. We found that a five unit hidden layer with a learning rate of 0.0001 and momentum of 0.1 worked well.

3 Results on Actual LADAR Data

As described in the previous section, the networks were trained with synthetic imagery. The networks were the tested on 15 real LADAR images[1] from the Fort Carson data set.

When the exhaustive template probing was used, it found the correct vehicle in 11 of the 15 images examined. We then looked at the number of times the network prediction failed to predict the top 25 correct templates. Out of the 375 (25 templates × 15 images) templates, the network prediction failed only 14 times for a success rate of about 96%, while still achieving the same identification rate.

The neural network selection of salient templates decreases processing time. Figure 1 shows the run-times for all chips examined in the 15 images. Figure 1a shows the cumulative time, and Figure 1b shows the time in seconds for each window using the two approaches. The experiments were run simultaneously on a Sparc 20. The drop in time per window for the exhaustive method corresponds to the completion of the network indexing method.



a. Cumulative time **b.** Time per window

Figure 1: Execution times.

References

[Beveridge *et al.*, 1994a] J.R. Beveridge, A. Hanson, and D. Panda. Integrated Color CCD, FLIR & LADAR Based Object Modeling and Recognition. Tech Report, Colorado State University, 1994.

[Bevington, 1992] J.E. Bevington. Laser Radar ATR Algorithms: Phase III Final Report. Tech Report, Alliant Techsystems, Inc., 1992.

[Fang *et al.*, 1987] Z. Fang, X. Li, and L.M. Ni. Parallel algorithms for image template matching on hypercube SIMD computers. *PAMI*, 9(6), 1987.

[Katz *et al.*, 1993] A. Katz and P. Thrift. Hybrid neural network classifiers for automatic target detection. *Expert Systems*, 10(4), 1993.

[Stevens *et al.*, 1997] M.R. Stevens, C.W. Anderson and J.R. Beveridge. Efficient Indexing for Object Recognition Using Large Networks. *ICNN*, 1997.

[Perlovsky *et al.*, 1995] L.I. Perlovsky, J.A. Chernick, and W.H. Schoendorf. Multisensor ATR and Identification of Friend or Foe using MLANS. *Neural Networks*, 8(7-8), 1995.

[Rumelhart *et al.*, 1986] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing, Volume 1: Foundations*, The MIT Press, 1986.

[1] The Fort Carson data collection is publicly at: `http://www.cs.colostate.edu/~vision`

Evaluations of Large, Complex Research and Development Programs: Theory and Practice

Theodore R. Yachik and Lynne Gilfillan
LGA, Inc.
12500 Fair Lakes Circle, Fairfax, VA
yachikth@lga-inc.com, lgilfill@lga-inc.com
www.lga-inc.com

Abstract

Evaluation of large, complex, research and development programs requires a new, and more structured approach. The approach developed and used here is an evaluation framework whose evaluation elements are based on detailed consensus-based evaluation plans. Implementation of this approach is illustrated in a description of the evaluations designed and performed by the authors for the Defense Advanced Research Projects Agency Unmanned Ground Vehicle (DARPA/UGV) program. Current data is presented on the evaluations of the UMASS color detection and the Hughes FLIR detection algorithms for the Ft. Carson collection.

Implementation of the evaluation approach for UGV/RSTA[1]

Background

The RSTA module of the UGV program was developed around the assumption of the use of three sensor modalities: Color CCD, 8-14 micron FLIR, and LADAR. The operational concept prescribed a two stage approach, detection followed by identification. The approach was to encompass many performers who were to develop algorithms for these specific sensors or for a combination of sensors. In general, the detection would be performed by either a Color CCD sensor or with a FLIR. The detection result would be passed to a different algorithm that would use the FLIR or LADAR sensors, either singly or in combination to perform identification.

The initial sensor design had fixed a number of the important characteristics and operational conditions in a manner that had a substantial impact on the program and ultimately the evaluations. The most critical of these decisions was to develop a LADAR sensor specifically for this program. The sensor was not built in time to provide any training, testing or evaluation imagery, thereby making it impossible to perform any formal evaluation of algorithms based on LADAR imagery. The second critical design decision concerned the focal lengths of the lenses for the Color and FLIR sensors. The design philosophy was to use a wide field of view (11.11 degrees) lens for detection and a narrow field of view lens (1.23 degrees) for identification. These design constraints, with the focal plane pixel size, spe-

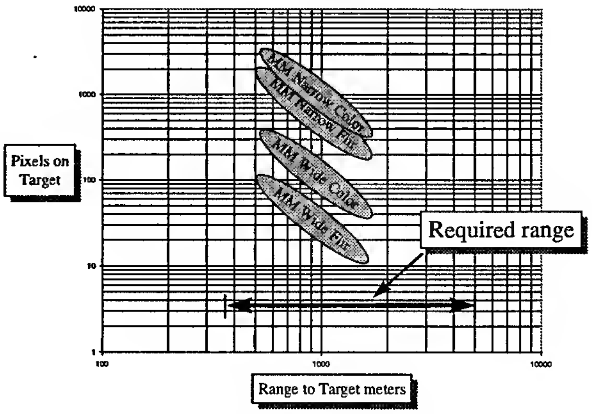


Figure 1 RSTA POT vs. RTT

[1] This work was sponsored by the Defense Advanced Research Projects Agency under contract number DAAH01-93-CR113 and monitored by the U. S. ARMY Missile Command. It is an SBIR effort.

cific targets, and the operational ranges to target resulted in a design that produced a specific pixels on target verses range-to-target relationship (see Figure 1). At the anticipated maximum range-to-target of 5,000 meters, there would only be 3 pixels on target for detection. It is within this framework that the algorithms had to be evaluated.

Developer Algorithm descriptions

The following sections describe the major types of evaluation experiments in which performers participated, the technical approach, and the partition of mission requirements addressed.

Color CCD detection

Only one performer produced a color CCD detection algorithm for UGV/RSTA. This algorithm was developed by the Colorado State University/ University of Massachusetts team. The algorithm identifies targets based on the differences in hue between targets and background. Since the sensor operates in the visible portion of the spectrum, detections using this algorithm could be carried out during the day only, even though mission performance requirements specify both day and night detection. Thus a major mission performance shortfall was identified prior to any "actual" evaluations.

FILR detection

Several performers developed algorithms using FLIR imagery to perform detection including Lockheed-Martin Marietta, Hughes, and Honeywell. Their FLIR algorithms were designed to take as input a full-frame, wide-field-of-view image. The algorithms then detected potential targets and passed the locations of these targets and small image chips containing these targets to the next processing stage. Currently, only the Hughes algorithm has been evaluated.

Color CCD Evaluation

Objectives

There were two primary objectives for the evaluation of the University of Massachusetts (UMASS)/Colorado State University CCD detection algorithm: (1) to determine the range of user parameters where the algorithm exceeded the user requirements and (2) to determine the performance of the algorithm over the selected data set in terms probability of detection and false alarm rate. Ancillary objectives were to provide an analysis of performance to the program manager and the developer to support identification of specific problems with either the algorithm or the data.

Implementation details

The evaluation of the UMASS/Colorado State University color detection algorithm was performed at the University of Massachusetts under the supervision of the evaluation team. University of Massachusetts staff supporting the conduct of the evaluation had no previous knowledge of the specific images selected for the evaluation. The evaluation design matrix contains both sequestered and non-sequestered imagery.

The sequestered imagery is from the Martin Marietta September 94 collection (Rimey, 1995) and consists of approximately 111 unique image scenes. There were 199 target opportunities in the unique image scenes. Each unique image scene in this data set had 10 or more replicate images[2]. Therefore, in the sequestered evaluation sample, there were 1990 target opportunities.

The entire Ft. Carson set (Beveridge, 1994) had been released to the developers early in the program for algorithm development and therefore was not sequestered. The Ft. Carson collection consists of scanned color film, with 37 unique image scenes with 56 targets. There were no replicate images for this collection.

The non-sequestered Ft. Carson data set was used in the evaluation because it contains image acquisition conditions significantly different from those in the Martin Marietta collection. It therefore offered an opportunity for conducting an evaluation over conditions closer to the mission requirements than would have been possible had we used only

[2] Replicate images are "duplicates" of the unique image scenes, captured at the frame rate of the camera (e.g., 1/30 sec). For this collection 10 replicate images were captured for each unique image scene. The images should be identical except for variations caused by system noise and camera vibration.

the sequestered images from the Martin Marietta 94 collection. We believed that this opportunity for breadth in evaluation outweighed the fact that developers had had the opportunity to train on the test set -- not usually a desirable evaluation condition.

One significant difference between the Ft. Carson and the Martin Marietta collections is that the Ft. Carson data was obtained from digitally scanning film images, while the Martin Marietta collection captured the images directly from a digital CCD camera. The scanned film images were processed and reduced in resolution by a factor of 4 using the Kodak Photo CD process (this process first spatially low passes the image before subsampling to reduce aliasing) to 490 by 760 pixels. The film images had variations in exposure and the best exposure images available were chosen for the evaluation. The CCD images were collected with an automatic "white balance" control operating. This control normalizes the gain of each color to provide an overall "best exposure." However this "white balance," because it adjusts each color on an image-by-image basis based on the scene content, destroys the absolute color information. In images where there is a large amount of sky in the image, there is a noticeable color shift in the background.

No additional development on the version of the algorithm used for evaluation was conducted by the UMASS/Colorado State University after November 1995. For the evaluation, the algorithm was to have two sets of operating characteristics, one developed for each of the two collections. The algorithm would use one set of parameters for all of the acquisitions from each collection. In this manner, the evaluation would characterize the variance in the algorithm's ability to perform over different days and over different scenes within a restricted time and location (6 days at Ft. Carson, 4 days at the Martin Marietta collections). Information was passed to the algorithm that identified the collection set, but no additional information, (e.g., illumination conditions, color normalization, target type, or range-to-target) was used by the algorithm.

Conditions of evaluation

Target set

For the best evaluation results, targets should be equally distributed over the target types and target orientation. (The design assumes a stratified sample). This was not the case for this sample, even after both collections were combined. The distribution of targets and their orientations is not uniform -- almost half as many targets are oriented front/rear as are diagonal or broadside. Thus, for any given range, there will be more POT for the set of targets than if the poses had been uniformly distributed. This should make detection easier than it would be in an entirely random selection. In addition there are too few targets (by a factor of two) of the M543A2 and M113-901 type as there should be (based on a uniform distribution of targets).

Image set

A summary table of the image data set characteristics that were chosen for evaluation are in Figure 1. The pixels on target for the Ft. Carson collection ranged from 500 to 4000, while the range for the Martin Marietta collection was 30 to 4000. In addition the Ft. Carson collection was done at very low ranges to target (50 to 180 m) while the Martin Marietta collection was at substantially greater ranges to target (500 to 1600m). The range-to-target is significantly greater in the Martin Marietta collection and much more comparable to the expected operating conditions.

Though the maximum pixels on target is approximately the same for each collection (4000), the images in the Martin Marietta collection have a much greater range of pixels on target available, mainly as a result of using an additional wide field-of-view lens. Any atmospheric effects that would effect the color would more than likely be seen in this image set.

Table 1 shows both the image-science oriented and the user-oriented (i.e., subjective scale data) descriptive data on the images. All of the imagery was rated as having high signal-to-noise (no noise visible in the images). All of the images in the Ft. Carson imagery were judged to have high visibility and low environmental clutter, while the Martin Marietta collection images were judged to vary substantially on both dimensions.

Evaluation parameters and ranges

Target scene and sensor

The targets that were used in this evaluation were painted with standard army camouflage paint. All of the targets were stationary during imaging. The illumination conditions during the collections were day only and the weather was generally clear.

The background was generally treeless, green grassy fields with slow rolling hills at Ft. Carson. There were very few bushes and mostly unstructured backgrounds. Some of the imagery had mountains in the distant background, but in most of the imagery the horizon was close to the camera. Almost all of the images include significant portions of sky.

The backgrounds available at Martin Marietta are much more complicated than at Ft. Carson. Most of the imagery is of distant rolling brown grasslands, but there are a number of clumps of trees and even some large green bushes. There are images with very structured, large rock outcrops. There are some dirt roads and some buildings. A number of the narrow field of view images do not contain any portion of the sky.

The limited conditions of imagery collection precludes performance evaluations that are consistent with many mission requirements.

Scoring rubrics

Detections were scored on the basis of consistency with the available "ground truth." Ground truth on the images was produced by manually drawing a bounding rectangle around each *true* target on each image. No independent ground truth, based on actual conditions on the ground was available.

Targets were counted as detected if the center of the target report fell within the ground truth bounding rectangle. *True* targets that are in the image, but are not in the evaluation set (e.g., the pick-up truck) were not counted as either detections or misses or false alarms. Targets that were within 10 pixels of the edge of the frame, or partially off the frame were not counted. Multiple detections on the same target are reported as a single true detection. The number of multiple detections, however, is reported.

False Alarm Rates (FAR) were determined by counting all of the reported detections that are not true detections. This FAR was calculated on a frame by frame basis. True targets that are de-

Table 1 Color evaluation imagery characteristics

	Ft. Carson	**Martin Marietta September**
Description of imagery	Scanned color film	CCD camera auto "white balanced"
Number of replicates	Single replicate	10 replicates/scene
Number of unique images	37	110
Number of targets	56	199
POT	4000 to 500	4000 to 30
Range-to-target	50-180 meters	500 to 1600 meters
Target visibility	100% high visibility	74% high, 19% medium, and 7% low visibility
Obstruction	54% none, 43% minor, 5% major	46% none, 38% minor, 17% major
Visible signal-to-noise	100% high	100% high
Environmental clutter	100% low clutter	20% low, 48% medium, and 32% high clutter
Sequestered/Not-sequestered	not sequestered	sequestered

tected were not be counted in the FAR. Multiple detections within a single "true target" bounding rectangle were not counted in the FAR.

A second analysis was performed treating the pick-up truck as a true target, even though it was not in the original list of targets. This was done because it was a camouflaged military pick-up truck, and it seemed reasonable to look at the results in this framework as well.

To develop performance requirements, we worked with the University of Texas at Arlington using their Scenario Based Engineering Process, and with the algorithm developers. We were unable to determine any specific requirements for the probability of detection (P_d) or the False alarm Rate (FAR), but we were able to come to a consensus based on reasonability of P_d>= 0.9 and less than 1 false alarm per 60 frames as performance goals.

Algorithm training

The instructions that the UMASS staff received on training their algorithm were consistent with the expected operational environment. The algorithm was to be trained in a manner that would allow separate parameter setups to be used for each collection. UMASS staff were allowed to select the specific samples to train on, including determining the content, range of parameters represented, and ultimately the number of images trained. This approach was used in order to allow the developers the optimum flexibility in selecting an operating point that could trade off Probability of Detection (P_d) and FAR.

UMASS staff successfully trained their algorithm on the Ft. Carson imagery, but were unable to train their algorithm on the Martin Marietta collection imagery. They identified two problems with the Martin Marietta collection imagery that prevented them from successfully training the algorithm. In the first image of the ten image replicate set, there was a shift in the mean response in the blue band of 18 counts (c.f. Table 2) which translates to almost 180° hue shift. The second problem that UMASS staff identified is that between images, the hue of the background undergoes large shifts. Since the algorithm uses the difference in hue between the targets and the background, these hue shifts made the algorithm un-trainable for the Martin Marietta collections.

Table 2 Mean band response for two Martin Marietta collection images

Mean Responses	**Red Response**	**Green Response**	**Blue Response**
Frame 1	167.9	167.2	142.1
Frame 2	166.8	165.0	**170.5**

The problems experienced by the UMASS staff in training on the Martin Marietta color images are believed to be the result of the following two circumstances:

1. The CCD camera used in the Martin Marietta collection was operated with the automatic white balance control functioning. This resulted a loss in absolute color across images, since the color is adjusted to the overall color of each scene. Since the UMASS/Colorado State University algorithm is based on the absolute colors of both targets and background, the image to image variation in the CCD images caused serious problems with algorithm performance.
2. The background vegetation in the Martin Marietta collection was gray/brown, resulting in an indeterminate hue. Since the algorithm makes distinctions based on color hue, the indeterminacy of the hue caused serious problems with algorithm performance.

We were unable to correct for these effects. Therefore, although we had planned to use both the Martin Marietta and the Ft. Carson imagery sets, the evaluation of the UMASS/Colorado State University algorithm was actually performed using the Ft. Carson imagery only.

Evaluation imagery selection and specification

The UMASS/Colorado State University algorithm was tested on 37 images from the Ft. Carson collection. The major characteristics with respect to POT of the 56 targets contained in the 37 images

Table 3 Summary target information for color test imagery

Target	Pose	POT(>2000)	POT (1000-2000)	POT (250-1000)	Grand Total
M60	Front	0	4	0	4
	Diag	0	3	2	5
	Broad	8	1	0	9
M60 Total		8	8	2	18
M113	Diag	4	3	1	8
	Broad	0	2	9	11
M113 Total		4	5	10	19
M113-901	Front	0	0	1	1
	Diag	0	3	1	4
	Broad	2	9	3	14
M113-901 Total		2	12	5	19
Grand Total		14	25	17	56

are shown in Table 3. There are only three target types, and one of them, the M113-901, is a minor modification of another target type. There are essentially no images with the M113 oriented head-on to the sensor. All images contain at least one target. Most importantly, there are no targets with fewer than 250 pixels on target. The limited amount of useable evaluation imagery available seriously compromised the utility of this evaluation, particularly in terms of end-user mission objectives.

Results

UMASS staff ran their algorithm on 37 images and reported the bounding rectangles for each detection. Targets with bounding rectangles that were within 10 pixels of the edge of the frame, or partially off the frame were not counted. On two occasions multiple detections on the same target were counted as a single true detection.

Table 4 Color algorithm evaluation summary results

	Probability of Detection, per target	**False Alarm Rate, per frame**
Performance Goals	90%	0.0166
Algorithm performance	87.4% (+/- 10%)	6.7 (+/- 1.8)

Performance results are displayed in Table 4. The P_d (87.4% +/- 10% $_{\text{95\% Confidence Interval}}$) is not different from the end-user performance goals of 90% P_d, at statistically significant levels. However, this is partially a result of the relatively small sample size and the resulting large confidence interval. It is possible that the algorithm parameters could be tuned to increase the detection rate, but at the expense of increasing the false alarm rate.

There was a substantial false alarm rate. On average there were 6.7 false alarms per image with 95% of the images having between 0 and 18 false alarms per image

Almost all of the images (33 out of 37) had at least one false alarm and four images had 15 or more false alarms. The average false alarm rate is 400 times greater than the end- performance goals of one false alarm per 60 frames.

The impact of the false alarm rate depends on the details of the operational scenario. If **only** the color detection algorithm were to be used, it is probable that detected targets would be chipped out and transmitted to a base station for display and evaluation by a human observer. In this scenario, the algorithm would function as an image pre-screener and missed targets would have a substantial impact on utility. The false alarm rate would impact the operational performance in both time and bandwidth requirements to send back each detected target and enough surround for context. Given the limited bandwidth of the link to the ground-station, the impact of the large number of false alarms is substantial in the amount of time that the scout must remain exposed.

Table 5 Summary of color false alarm types

False Alarm Type	Percent
Target like	6.8%
On horizon	18.9%
On bushes on horizon	12.8%
On near field bushes	11.5%
In sky	28%
Other	22%

Six types of false alarms were identified in the imagery (see Table 5). A camouflage-painted pick-up truck was in 14 images and was counted in the first analysis as a false alarm since it was not in the original target set. If one were to consider the camouflage-painted truck that was imaged as a target, rather than a false alarm, there is no statistically significant change in the detection and false alarm rates. The probability of detection increases from 87% to 90% (+/- 10%) and the false alarm rate drops from 6.7 to 6.1 (+/- 1.8) per frame.

Almost 32% (18.9% + 12.8%) of the false alarms occurred along the line of the horizon, either where a tree or bush protruded above the horizon or just adjacent to the horizon line itself. Detections within bushes and other scrub that were in the near field of view, close to the camera contributed 11.5% of the false alarms. Detections in the sky, often five or ten within a single image, contributed 28% of the false alarms. The rest of the false alarms came from a variety of undifferentiatable causes.

Even though algorithm detections that were within 10 pixels of the edge of the image frame were removed from the analysis, there were a significant number of these detections (3.4 (0, 11) 95% detections per frame). The implications in an operational system are an increase in algorithm complexity and processing time to extract the reduced size image. A more serious impact is the 4% reduction in effective image size that would require additional image-to-image overlap when creating mosaics.

FLIR Evaluation

Objectives

There were three primary objectives for the evaluation of the Hughes FLIR detection algorithm: (1) to determine the range of user parameters where the detection algorithm met or failed to meet the user requirements, (2) to determine the performance of the algorithm over the selected data set in terms of probability of detection and false alarm rate, and (3) to evaluate the hypothesis that an adaptive algorithmic approach is superior to a non-adaptive approach.

Implementation details

The evaluation of the Hughes FLIR detection algorithm was performed at NVESD under the supervision of the evaluation team. The evaluation imagery was selected as described previously and the specific images selected were not known to the Hughes development team until the completion of the evaluation. The initial evaluation, which is all that is reported here, includes imagery from the Ft. Carson collection only. Images from the Ft. A.P. Hill and Martin Marietta September collections are currently being processed. The evaluation design matrix consists of 84 targets contained in 58 unique images collected over six days. All of these images were available to the developer. There were no sequestered or replicate images used in this evaluation.

No additional development on the version of the algorithm used for evaluation was conducted by Hughes after September 1996. Hughes developed two sets of parameters specifically for the Ft. Carson collection. One parameter set is for a non-adaptive algorithm, while the second set is for an adaptive algorithm. Information was passed to the algorithm that identified the collection set, but no additional information, (e.g., illumination conditions, time of day, target type, or range-to-target) was used by the algorithm. It is anticipated that the continuing testing will utilize four additional parameter sets, two for each of the two other collections.

Conditions of evaluation

Target set

As discussed in the color evaluation section, it is desirable to have a uniform distribution of samples in each sample strata factors. However, the distribution of targets and orientations for this evaluation is not uniform -- almost one-third as many targets are oriented front/rear as are diagonal or broadside. For this target set, at a specific range, the number of pixels on target is greater for broadside and diagonal orientations than for the front/rear orientation. The distribution over the three target types is roughly uniform; however, there are fewer M113-901 oriented broadside, and more oriented diagonally, than one would expect.

The original evaluation memorandum of understanding only considered three vehicles as "true" targets: the M60, M113, and M113-901. In the Ft. Carson collection there are a number of images of a camouflaged painted pickup truck. Two separate analysis were performed, the first used only the targets initially agreed to as targets, the second included the pickup truck in the target set. The pickup truck provides an additional 24 targets, mostly in a diagonal orientation (2 front/rear, 16 diagonal, and 6 broadside).

Image set

A summary table of the image data set characteristics chosen for this evaluation are shown in Table 6. The pixels on target for the FLIR Ft. Carson collection ranged from 150 to 1900, significantly less than the POT used in UMASS evaluation (500 to 4000). The RTT is the same as used in the color evaluation and as such is significantly fewer than the operational requirements. This implies that the atmospheric effects are not consistent with what would be expected operationally, even in those cases where the POTs are appropriate.

Table 6 shows both the image-science oriented and the user-oriented (i.e., subjective scale data) descriptive data on the images. The FLIR imagery was rated as generally medium visibility (44%), that is between 25% and 75% of the boundary of the targets were visible. The imagery was judged to have high Visible-Signal-to-Noise (VSNR) (84%) but there are a number of images with VSNR in the medium and low range. About half (45%) of the targets are obstructed.

Evaluation parameters and ranges

Target scene and sensor

The targets that were used in this evaluation were painted with standard army camouflage paint. All of the targets were stationary during imaging. The illumination conditions during the collections were mostly daytime with only three images acquired at night. The weather was generally clear. The background was generally treeless with green grassy fields and slow rolling hills. There were very few bushes and mostly unstructured backgrounds. Some of the imagery had mountains in the distant background, but in most of the imagery the horizon was close to the camera. Almost all of the images include significant portions of sky. In most cases the target engines were running and the target bodies were hot.

As with the color evaluation, the limited conditions of imagery collection precludes performance evaluations that are consistent with many mission requirements.

Scoring rubrics

The scoring rubrics used for the FLIR evaluation were the same as used in the color evaluation. Detections were scored on the basis of consistency with the available "ground truth." Ground truth on the images was produced by manually drawing a bounding rectangle around each *true* target on each image. No independent ground truth, based on actual conditions on the ground, was available.

Table 6 FLIR evaluation imagery characteristics

	Ft. Carson FLIR Imagery
Description of imagery	Amber FLIR
Number of replicates	Single replicate
Number of images	58
Number of targets	84 (109 including pickup truck)
POT	1900 to 150
Range-to-target (RTT)	50-180 meters
Target visibility	27% high, 44% medium , 29% low visibility
Visible signal-to-noise (VSNR)	84% high, 15% medium, 1% low signal-to-noise (S/N)
Obstructed	55% none, 30% minor, 15% major obstruction
Environmental clutter	1% high, 65% medium, 34% low clutter
Sequestered/ Non-sequestered	not sequestered

Targets were counted as detected if the center of the target report fell within the ground truth bounding rectangle. *True* targets that are in the image, but are not in the evaluation set (e.g., the pick-up truck) were not counted as either detections or misses or false alarms. Targets that were within 10 pixels of the edge of the frame, or partially off the frame were not counted. Multiple detections on the same target are reported as a single true detection. The number of multiple detections is reported.

False Alarm Rates (FAR) were determined by counting all of the reported detections that were not true detections. The FAR was calculated on a frame by frame basis. True targets that are detected were not counted in the FAR. Multiple detections within a single "true target" bounding rectangle were not counted in the FAR.

A second analysis was performed treating the pick-up truck as a true target, even though it was not in the original list of targets. This was done because it was a camouflaged military pick-up truck and it seemed reasonable to look at the results in this framework as well.

Algorithm training

The instructions that the Hughes development team were given on training their algorithm were consistent with the expected operational environment. The algorithm was to be trained in a manner that would allow separate parameter set-ups to be used for each collection. Hughes was allowed to select the specific samples to train on, including determining the content, range of parameters represented, and ultimately the number of images trained. This approach was chosen to allow the developers the optimum flexibility in selecting an operating point that could trade off P_d and FAR.

Hughes successfully trained their algorithm on the Ft. Carson imagery and delivered two sets of parameters, one non-adaptive and one adaptive, to NVESD for testing. Hughes did not choose to develop separate parameters for day and night imaging; however, they did choose to develop separate parameter sets for each collection (only the Ft. Carson set was tested at the same time as publication).

Evaluation imagery selection and specification

The imagery that was used for the Hughes detection evaluation was the entire set of Ft. Carson imagery that was selected, as described in Table 6. No images were set aside, and the entire evaluation was performed in one algorithm execution session.

The Hughes FLIR detection algorithm was tested on 58 unique images, containing 84 targets. The majority (84%) of the targets have POT between 250 and 1000. This is significantly fewer than the POT used in the color evaluation (c.f. Table 1). As with the color evaluation, the limited amount of useable evaluation imagery available seriously compromised the utility of this evaluation, particularly in terms of end-user mission objectives.

The algorithm scoring was provided by the NVESD AUTOSPEC program (Walters, 1990) which automatically scores detections, misses, and false alarms using the rules detailed in the scoring rubrics.

Results

NVESD ran the Hughes algorithm twice on all of the 58 Ft. Carson images and scored the results. The Hughes algorithm only reported a single point (centroid) for each detection. There were no targets reported within 10 pixels of the edge of the frame and there were no multiple detections on the same target.

The performance results are displayed in Table 7. The P_d for both the non-adaptive (59% +/- 10% $_{\text{95\% Confidence Interval}}$) and adaptive (49% +/- 10% $_{\text{95\% Confidence Interval}}$) cases are different from the end-user performance goals of 90% P_d at statistically significant levels. The adaptive P_d is not different than the non-adaptive P_d, at statistically significant levels. However, this is partially a result of the relatively small sample size and the resulting large confidence interval.

There was a substantial false alarm rate. On average there were 4.7 false alarms per image with the non-adaptive algorithm. The false alarm rate decreased, at statistically significant levels, to 1.4 false alarms per image, when using the adaptive algorithm. The false alarm rate for the adaptive algorithm is 84 times the end-user performance goal of one false alarm per 60 frames.

The use of the adaptive FLIR algorithm improves the FAR, at the expense of the P_d. With a fixed decision process, when the decision threshold is changed, both the P_d and FAR change along a fixed Receiver Operator Characteristic (ROC) curve. This ROC fully determines the algorithm's

Table 7 FLIR algorithm evaluation summary results

	Probability of Detection (P_d), per target	**False Alarm Rate (FAR), per frame**
Performance Goals	90%	0.0166
Algorithm performance **non-adaptive**	59% (+/-10%)	4.7 (+/- 0.2)
Algorithm performance **adaptive**	49% (+/- 10%)	1.4 (+/- 0.4)

performance with respect to P_d and FAR.

Two algorithms are different in performance, if and only if their ROC's are different. Therefore it is important to determine if the measurements of P_d and FAR for these two algorithms lie on two different ROC curves. Unfortunately, we do not have enough data from this evaluation to answer this question. Additional information that is required to calculate the ROC curves is the confidence in each detection. This information is the intermediate data before a decision threshold is applied. With the confidence information, and additional data points to increase the measurement precision, one would be able to determine if the two algorithms were the same decision process, with different thresholds, or were two, differently performing, algorithms. The number of data points that would be necessary to measure a statistically significant difference of 10% in P_d (e.g., 90% and 80%) at the same FAR, would be approximately 350 (as compared to the 86 used).

Comparison between color and FLIR evaluation results

Both the color and FLIR evaluations were performed on imagery that was taken at the same collection. Even though there are obvious differences in the modalities of these two evaluations, it is useful to look at both their differences and their similarities. Table 8 details the image characteristics for each evaluation. The image sizes of the two sensors are very different with the color imagery 1.5 times as wide as the FLIR (40° vs. 25°). The resolution (IFOV) of the color sensor is 1.6 times as good as the FLIR sensor (1.6 mr. vs. 1.0 mr.). The average POT of the color imagery is almost four times the average POT of the FLIR imagery (1610 vs. 485). The implication is that the color imagery had significantly more pixels on target with which to make a detection than the FLIR.

Table 9 compares the performance of the algorithms using a number of metrics. The color detection meets the end user requirements of a 90% P_d, while neither the non-adaptive nor the adaptive FLIR algorithm does. The inclusion of the pickup truck in the target set does not change any of the performance results. The P_d is about 90% for the

Table 8 Comparison of color and FLIR image characteristics

	Color evaluation	**FLIR evaluation**
Description of imagery	Scanned color film	3.5 micron AMBER FLIR
Field of View (FOV)	40.4 ° (H), 28.4 ° (V)	24.9 ° (H), 23 ° (V)
Image size	768 x 512pixels	256 x 256 pixels
IFOV	0.9 x 1.0 mr.	1.7 x 1.6 mr.
Number of replicates	Single replicate	Single replicate
Number of unique images	37	58
Number of targets	56	86
POT	500 - 4000 (mean 1610)	150 - 1900 (mean 485)
Range-to-target	50-180 meters (mean 108)	50 to 180 meters (mean 114)
Target visibility	100% high visibility	27% high, 44% medium , 29% low visibility
Obstruction	54% none, 43% minor, 5% major	55% none, 30% minor, 15% major obstruction
Visible signal-to-noise	100% high	84% high, 15% medium, 1% low S/N
Environmental clutter	100% low clutter	1% high, 65% medium, 34% low clutter
Sequestered/Not-sequestered	not sequestered	not sequestered

Table 9 Performance comparisons of color and FLIR evaluations

	Color Detection		FLIR Detection			
			Non-Adaptive		Adaptive	
	Pickup Truck False Alarm	Pickup Truck Target	Pickup Truck False Alarm	Pickup Truck Target	Pickup Truck False Alarm	Pickup Truck Target
Pd	87% (+/-10%)	90%(+/-10%)	59%(+/-10%)	63%(+/-10%)	49%(+/-10%)	54%(+/-10%)
FA/Frame	6.7 (+/- 1.8)	6.1(+/- 1.8)	4.7 (+/-0.2)	4.4(+/-0.2)	1.4 (+/- 0.4)	1.1(+/- 0.4)
FA/Degree	0.006	0.005	0.008	0.008	0.002	0.002
Targets/ Reported detections	18%	24%	16%	21%	34%	48%
Frames with out any FA	11%	11%	0%	0%	29%	55%
Frames > 5 FA	62%	62%	66%	52%	7%	7%

color algorithm and about 60% for the FLIR non-adaptive and 50% for the FLIR adaptive algorithm. There are about 6.5 False alarms per frame with the color algorithm, and 4.5 and 1.3 false alarms per frame with the FLIR non-adaptive and adaptive algorithms respectively. None of the algorithms meet the end user requirements for FAR of 1 per 60 images.

Since the color and FLIR imagery have different sizes and IFOVs, it is useful to compare their false alarm rates on comparable basis. When the data is normalized by square degrees, one comes to a different conclusion on the relative false alarm rates of the algorithms. The non-adaptive FLIR algorithm has the highest FAR (0.008), followed by the color (0.005), and then by the FLIR adaptive algorithm (0.002). These are the metrics that an end-user is concerned with, since they relate to the expected number of false alarms that they would see when they are performing surveillance on a piece of terrain. For example, viewing a 45 degree wide, 20 degree high area, one would expect 7 false alarms using the FLIR Non-adaptive algorithm (45x20x0.008).

Another metric of importance to the end-user is how many detections they must review until they find a target. This is measured by the percentage of detected targets to reported detections. Only about 20% of the detections that the color and the non-adaptive FLIR algorithms report are "true" targets. The adaptive FLIR algorithm does somewhat better, increasing to 48% when the pick up truck is considered as a target. However, even at best, one half of the detections are targets, and one misses one half of the targets.

The distribution of the number of false alarms per frame is not the same between the color and FLIR algorithms. The FLIR non-adaptive algorithm produces false alarms in every image, but the adaptive algorithm (with the pick-up truck) produced no false alarms in 50% of the images. In the color algorithm 11% of the images have at least one false alarm. More importantly is the distribution in the number of false alarms per frame. About 60% of the color and non-adaptive FLIR algorithm processed images have more than 5 false alarms per frame, as compared to the 7% of the adaptive FLIR processed images. In the first case, the false alarms are bursty on an image by image basis. If there is one false alarm, there tend to be many false alarms.

Discussion

The development of the evaluation framework, the evaluation design, the implementation of the evaluation, and the evaluation results raise a number of important issues: (1) None of the algorithms met the end-user performance requirements developed by the SEP process, (2) The imagery data available for analysis was deficient, which did not permit the evaluation of algorithm performance over the range of end-user specified performance conditions, and 3) The algorithm evaluation process did not collect all of the data that is necessary to provide complete analysis.

All of the algorithms have been evaluated over a common space that is small with respect to user requirements. Five performance objectives were identified: probability of detection, false alarm rate, algorithm execution speed, location error, and degree of identification. However, the algorithms were evaluated using only three of these performance measures: P_d, FAR, and location error. Even over this limited set of evaluation conditions and performance measures, only the color algorithm meets the user goals and only with respect to P_d. No algorithm has false alarm rates that approach the user goals. In the best case, the false alarm rate is 66 times the goal. Additionally, it should be noted that the color algorithm is not effective at night. This level of performance implies that significant improvements in the algorithms' performance will be required before they are suitable for the Army scout tasks.

An algorithm's performance, particularly the P_d are critically effected by system design factors outside of the algorithm designer's control. The pixels on target is a major factor in the P_d for a given algorithm. POT is determined mainly by user requirements (range to target and target type) and system design (focal length and array pixel size). In this evaluation, the combination of user requirements and system design resulted in images with few POT (cf *Figure 1*). In this evaluation the average POT were low (*Table 8*) with means of 1600 and 500 for the color and FLIR imagery, and minimum POTs of 500 and 150.

The ability to determine the performance of a RSTA algorithm is critically dependent on the availability of test imagery with the proper characteristics. There are two fundamental requirements for this test imagery. It must cover the requirements space, and it must provide sufficient number of independent samples to support statistical hypothesis testing. In practical terms, the test imagery must be acquired at the conditions that the operational algorithm will experience. There must be images with the same targets, at the same ranges, with the same configurations and articulations, with the same backgrounds, and the same atmospherics. Secondly, there can not be just one image at a test condition, but there must be multiple independent samples.

The collections of imagery available for this evaluation have always had serious deficiencies in terms of the extent to which there is coverage of the parameter space. These deficiencies have limited the ability of this evaluation to obtain its goal of determining the algorithm's ability to perform the user specified mission. The data bases necessary for this type of evaluation require careful and concentrated attention from the program inception. Sufficient funds must be made available to either collect operational imagery or to generate synthetic imagery. The evaluation of pilot collections early in the program should be a guide to full-scale collections.

Finally, the comparison of the color and FLIR evaluations raise a number of interesting questions that we cannot answer with the available data. Does the color algorithm perform differently (better) than either of the FLIR algorithms? Does the FLIR adaptive algorithm perform differently than the FLIR non-adaptive algorithm? These questions require the generation and comparison of ROC curves for the various algorithms. The RSTA algorithms were not designed to provide the necessary data for this type of analysis. Such data-collection mechanisms are reasonably easy to implement and would provide valuable data for guiding future research.

References

Beveridge, R. et al. *November 1993 Fort Carson RSTA Data Collection Final Report*, DARPA Contract DAAH04-93-G-442, January, 1994.

Doria, D., and D. Huttenlocher. Progress on the Fast Adaptive Target Detection Program. In *Proceedings of the Image Understanding Workshop*, Morgan Kaufmann, 1996.

Fleisher, J. et al. Guidelines for the Evaluation of Image Understanding Systems. In *Proceedings of the Image Understanding Workshop*, Morgan Kaufmann, 1992.

Gilfillan, L. et al. *Image Understanding Evaluation Metrics and Methods*, Final Report, DARPA Contract DAAH01-92-CR186, December, 1992.

Gilfillan L. and Yachik T. User-Centered Evaluation (UCE) Methods and Metrics. In *Proceedings of the Image Understanding Workshop*, Morgan Kaufmann, 1994.

Mettala, E. et al. Scenario-based Engineering Process for Reconnaissance, Surveillance, and Target Acquisition. In *Proceedings of the Image Understanding Workshop*, Morgan Kaufmann, 1994.

Night Vision and Electro-Optics Directorate. *Standard Analysis of Target Recognizers Part 1: Baselining Test Plan*, July, 1992, AMSEL-RD-NV-VMD-LET.

O'Keefe, R. et al. *Validating Expert System Performance*. IEEE Expert, May 1987.

Rimey, R. *Surrogate Semiautomonous Vehicle (SSV) RSTA September 1994 Data Collection* Final Report, DARPA Contract DAAH01-92-C-R1010, January 1995.

Sadjadi, F. A Perspective on ATR Technology, Signal and Image Processing Systems Performance Evaluation. In *Proceedings of Signal and Image Processing Systems Performance Evaluation.* SPIE Vol. 1310, 1990.

Walters, C. Removing the ATR Performance Evaluation Bottleneck: The C2NVEO AUTOSPEC Facility. In *Proceedings of Signal and Image Processing Systems Performance Evaluation.* SPIE Vol. 1310, 1990.

Weiner, B. *Statistical Principles in Experimental Design.* McGraw-Hill, 1971.

Yachik, T. DARPA Briefing. In *Proceedings of the Image Understanding Workshop*, Morgan Kaufmann, 1994.

SECTION VI
MULTIDICIPLINARY UNIVERSITY RESEARCH INITIATIVE
(MURI)

MULTIDICIPLINARY UNIVERSITY RESEARCH INITIATIVE (MURI)

PRINCIPAL INVESTIGATOR REPORTS

A Trainable Modular Vision System

R. Brooks[1], E. Grimson[1], T. Poggio[1], C. Koch[2], C. Sodini[1], L. Stein[1], W. Yang[3]

[1]MIT, Cambridge MA 02139
[2]Caltech, Pasadena CA
[3]Harvard University, Cambridge MA

Abstract

The focus of this MURI project is on trainable, modular, vision systems, that is, systems that can be easily reconfigured for a range of visual tasks, and easily reprogrammed through training on example images. The main subprojects include: a humanoid robot that serves as a platform for biologicaly motivated trainable visual modules, a computational framework for designing visual search engines for a range of tasks in image databases and video, and reconfigurable hardware specialized for trainable vision tasks.

1 Objectives

Our multi-institution project focuses on: 1) developing versatile vision and sensing systems that can be easily customized for different tasks in different domains, largely motivated by biological models of visual processing, 2) modeling and analyzing the capabilities of these systems, 3) estimating the system performance for tasks in robotics, inspection, medical diagnosis, surveillance and target recognition.

To reach these goals, a multidisciplinary approach – cutting across computer science, electrical engineering, statistics, neural networks, psychophysics and visual physiology – is key if we want to develop not simply another algorithm but a vision architecture that – like biological systems – is robust, flexible and easily adaptable to many different tasks.

Computer vision will be critical for advanced weapon systems, surveillance, transportation, medicine and manufacturing in the 21st century. Unfortunately, existing computer vision systems are still too inflexible for widespread use in military and commercial applications. Each machine perception task is different and typically requires very specific software and hardware. A small, generic vision-based system that could be easily customized without significant reprogramming will meet the forthcoming needs for reliable and autonomous performance.

Thus the central idea in our approach is to utilize a multidisciplinary team to develop *modular* and *trainable* architectures, inspired by *biological systems* and motivated by sound theories in signal processing, statistics, function approximation and neural networks. Modularity ensures that components may easily be upgraded or replaced with alternatives, or connected in alternative arrangements. Trainability ensures that the same general framework can easily be reused for different tasks, and can be programmed largely be exposure to examples.

Towards this end, we have built several prototypes that are *configured* from a set of basic software and hardware modules and *trained* to perform several different visual tasks such as image indexing, inspection tasks, reflexive visual behaviours for autonomous navigation, learn eye-hand coordination, classify scenes, and detect instances of object classes in images. Our project thus makes a step towards bringing together the disciplines of *machine perception* and *machine learning* and towards developing a system that can display some of the *robustness, flexibility and adaptability* of biological visual systems.

1.1 Specific components

The project's current foci include: a humanoid robot composed of reconfigurable, trainable, biologically motivated visual modules; a computational framework for creating visual search engines; and supporting projects in specialized hardware, low-level feature extraction and attention. Below, we highlight example successes and directions for future work.

[0]Research supported in part by ARPA and ONR under ONR contract N00014-95-1-0600. PIs may be contacted at brooks@ai.mit.edu, welg@ai.mit.edu, tp@ai.mit.edu. URL for project available at `http://www.ai.mit.edu/muri.html`

2 A Trainable Architecture for Object Classification and Detection

A central motivating example for much of our work has been that of designing search engines that can deal with large volumes of images, both in static databases and in video streams. To be effective at retrieving images in such cases, a search engine must be able to deal with classes of objects (e.g., find images with people in them, or with particular vehicles present), and with contextual information (e.g., find images of particular classes of scenes such as mountains, forests, cityscapes). This means we need a computational framework for efficiently representing the wide range of images that particular classes of objects can engender.

We are developing exactly such a new framework for synthesizing search engines for visual data bases of images and – in the future – video. Our goal is a system that can detect a wide range of images and objects in video and databases, including classes of objects such as faces, people, vehicles, and classes of scenes such as mountainscapes, fields, beaches and so on. The keys to such a framework are novel representations of images and their contents; flexible classifiers that can deal with large variations in objects and that can be trained efficiently from examples; and verification methods that can verify and screen detections raised by the classifiers.

2.1 Object Detection

In the past few years image-based techniques have achieved considerable success in synthesizing effective face detectors from a training set consisting of a large number of example face images. We developed such a system [28] which was also demonstrated for the similar task of detecting eyes. Rowley et al.[23] implemented a system with a similar architecture, that was significantly faster. More recently we have replaced the classifier used by [28] with a Support Vector Machine classifier [18] achieving almost real time performance on a Pentium platform. Attempts however to apply the same architecture to the detection of other types of objects encountered unexpected difficulties. It quickly became clear that the key problem was the specific image representation used. The Sung-Poggio system (and also the similar CMU system) use the simplest possible image representation, just pixel values. A pixel-based representation is sufficient in situations in which there is a reproducible pattern of grey values, e.g., frontal faces. It fails in tasks like the detection of people where the pixel pattern can be highly variable between different images of people. We have now found another representation which is biologically plausible and which seems to be much better for the difficult task of detecting people in complex images. It may well be a general represenation for many object detection tasks. The representation we propose for object detection consists of an object specific subset of an overcomplete dictionary of Haar wavelets.

The main contribution to finding a better representation than pixels is due to Sinha (1995) [25] who first suggested the use of *ratio templates* as a way to encode *qualitative* photometric and chromatic relations between image regions that are practically invariant under large changes of illumination and viewpoint. Lipson [14] has extended the scheme of Sinha and shown that it can deal successfully with the problem of natural scene classification.

2.1.1 Sinha's ratio templates

In his thesis [25], Sinha suggested representing objects as *ratio templates*, that is, sets of photometric inequalities between the average brightness values of pairs of appropriately chosen large image regions. The relations encode only the directions of the inequalities rather than their magnitudes. The image regions and their qualitative inter-relationships are encoded as a directed graph and the model-to-image matching process is handled as a graph-matching task. This template matching approach can be efficiently implemented in a multi-resolution framework, since the average brightnesses of different regions can be readily obtained by accessing single pixel values in the appropriate pyramid level. Sinha also devised a correlational learning scheme to extract qualitative object models from sets of normalized example images. The scheme was tested on the problem of detecting faces using a hand-crafted qualitative model for a human face comprised of about 15 inequality relations between image regions corresponding to the forehead, eyes, cheeks, nose, mouth and chin, achieving detection. As the experimental results demonstrate, these relations are robust to the photometric variations introduced in face images by changing illumination conditions. As a face detector the scheme did not, however, perform as well as Sung's or Osuna's, partly because of the limitations of the classification stage used by Sinha – a simple template matching.

2.1.2 Flexible templates for classification

Using this idea of relative relationships between image regions as a basis, Lipson has developed a framework for scene classification and object detection based on configural templates. Such templates again utilize relative relationships between image regions, including intensity, color, texture and other cues as bases. In a model template, extended regions are interrelated by flexible spatial relationships, which can be envisioned as a set of image regions connected by springs. Between pairs of regions one defines sets of relative relationships. A match of a model template to an image is measured by determining the extent to which such a flexible template must be stretched or twisted to fit the image, and the extent to which the relative relationships are found to hold between the corresponding image regions.

In initial testing, Lipson was able to show that hand crafted templates are very succesful at distinguishing classes of scenes, such as "snowy mountains" or "snowy mountains with lakes" or "fields" or "waterfalls" or "cityscapes", showing true positive rates on the order of 80% with false positive rates of only about 1%. This demonstrates the ability of the approach to deal with wide variations in color, intensity and spatial layout common within such large classes of scenes. Further, Lakshmi Ratan has demonstrated that by adding spatial information (in this case a Hausdorff matcher) within individual regions, one can easily extend the system to deal with detection problems, in this case detecting instances of images that contain cars.

Of course, hand crafted models are only a testbed. The goal is to create trainable systems, and we have built two such systems. The goal of each system is to allow a user to build his own template, by simply selecting a set of example images, and letting the system deduce a template that captures the commonality within the set of images. Although in many cases there is no unique common template for a set of images, we have demonstrated both approaches as successfully extracting templates that correctly retrieve the target class. Current work is focused on methods for including negative examples, methods to support refinement of templates by allowing the user to select examplars from retrieval sets, and methods of merging multiple cues into a common flexible template framework.

2.1.3 Image Representation: overcomplete dictionaries and sparsification

A general approach to object detection – being developed by Oren, Sinha, Girosi and Poggio – is to consider representations of images that can be customized for specific tasks. To this end, suppose we consider examples of the object class to be detected – like faces or people – resized to a standard size dictated by the type of object. Within this window, consider the dictionary of Haar wavelets. Two-dimensional Haar wavelets effectively encode bright-dark relations at various positions, scales and orientations in the image. They do so in a way that can be fully characterized mathematically. A complete set of Haar wavelets capture the full information in an image. In fact, we need an overcomplete dictionary because we need more dense shifts of the wavelets than strictly required by completeness (see Oren et al. in this volume). We could then simply analyze the image window in terms of this set of basis functions and provide the resulting coefficients as inputs to an appropriate classifier. The dictionary, however, is too large (say thousands of basis functions) to be a practical solution, since a high input dimensionality in the classifier almost always implies the need for a very large set of training examples.

For this reason, in our system we use a "learning" stage, in which a much sparser representaion is learned, based on the task. As shown in the paper by Oren et al. (this volume) a few example images of the object of interest are used to find which subset of the basis functions is needed to represent it, by keeping only those with non-zero absolute value of the coefficient, averaged over the training set. After this dimensionality reduction stage, only a few specific basis functions are needed to analyze the image and to provide the inputs to the classifier.

2.1.4 Scale and Position Invariance

As in the original architecture proposed by Sung and Poggio (and by many others before), detection is performed by moving the basic window used by the classifier across the image in space and in scale. Thus, scale and position invariance is achieved by brute force, simply by scanning and zooming an image. As described in Oren et al. (this volume) the scanning can be done somewhat more elegantly in the space of the wavelet coefficients.

2.1.5 The learning engine: Support Vector Machines

We use the Support Vector Machine (SVM), a pattern classification algorithm recently developed by V. Vapnik and his team at AT&T Bell Labs. [4, 6, 9, 32]. SVM can be seen as a new way to train polynomial, neural network, or Radial Basis Functions classifiers. While most of the techniques used to train the above mentioned classifiers are based on the idea of minimizing the training error, which is usually called *empirical risk*, SVMs operate on another induction principle, called *structural risk minimization*, which minimizes an upper bound on the generalization error. From the implementation point of view, training a SVM is equivalent to solving a linearly constrained Quadratic Programming (QP) problem in a number of variables equal to the number of data points. This problem is challenging when the size of the data set becomes larger than a few thousands. Osuna, et al., [17] have shown that a large scale QP problem of the type posed by SVM can be solved by a decomposition algorithm: the original problem is replaced by a sequence of smaller problems, that is proved to converge to the global optimum. The applicability of this approach was demonstrated by using SVM as the core classification algorithm in a real-time face detection system.

Here we briefly sketch the SVM algorithm and its motivation. A more detailed description of SVM can be found in [32] (chapter 5) and [9].

We start from the simple case of two linearly separable classes. We assume that we have a data set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ of labeled examples, where $y_i \in \{-1, 1\}$, and we wish to determine, among the infinite number of linear classifers that separate the data, which one will have the smallest generalization error. Intuitively, a good choice is the hyperplane that leaves the maximum margin between the two classes, where the margin is defined as the sum of the distances of the hyperplane from the closest point of the two classes.

If the two classes are non-separable we can still look for the hyperplane that maximizes the margin and that minimizes a quantity proportional to the number of misclassification errors. The trade off between margin and misclassification error is controlled by a positive constant C that has to be chosen beforehand. In this case it can be shown that the solution to this problem is a linear classifier $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{\ell} \lambda_i y_i \mathbf{x}^T \mathbf{x}_i + b)$ whose coefficents λ_i are the solution of the following QP problem:

$$\begin{array}{lll} \underset{\Lambda}{\text{Minimize}} & W(\Lambda) & = -\Lambda^T \mathbf{1} + \frac{1}{2}\Lambda^T D \Lambda \\ \text{subject to} & & \\ & \Lambda^T \mathbf{y} & = 0 \\ & \Lambda - C\mathbf{1} & \leq \mathbf{0} \\ & -\Lambda & \leq \mathbf{0} \end{array} \qquad (1)$$

where $(\Lambda)_i = \lambda_i$, $(\mathbf{1})_i = 1$ and $D_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$. It turns out that only a small number of coefficients λ_i are different from zero, and since every coefficient corresponds to a particular data point, this means that the solution is determined by the data points associated to the non-zero coefficients. These data points, that are called *support vectors*, are the only ones which are relevant for the solution of the problem: all the other data points could be deleted from the data set and the same solution would be obtained. Intuitively, the support vectors are the data points that lie at the border between the two classes. Their number is usually small, and Vapnik showed that it is proportional to the generalization error of the classifier.

Since it is unlikely that any real life problem can actually be solved by a linear classifier, the technique has to be extended in order to allow for non-linear decision surfaces. This is easily done by projecting the original set of variables $\mathbf{x}$ in a higher dimensional *feature space*: $\mathbf{x} \in R^d \Rightarrow \mathbf{z}(\mathbf{x}) \equiv (\phi_1(\mathbf{x}), \ldots, \phi_n(\mathbf{x})) \in R^n$ and by formulating the linear classification problem in the feature space. The solution will have the form $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{\ell} \lambda_i y_i \mathbf{z}^T(\mathbf{x})\mathbf{z}(\mathbf{x}_i) + b)$, and therefore will be nonlinear in the original input variables. One has to face at this point two problems: 1) the choice of the features $\phi_i(\mathbf{x})$, which should be done in a way that leads to a "rich" class of decision surfaces; 2) the computation of the scalar product $\mathbf{z}^T(\mathbf{x})\mathbf{z}(\mathbf{x}_i)$, which can be computationally prohibitive if the number of features n is very large (for example in the case in which one wants the feature space to span the set of polynomials in d variables the number of features n is exponential in d). A possible solution to this problems consists in letting n go to infinity and make the following choice:

$$\mathbf{z}(\mathbf{x}) \equiv (\sqrt{\alpha_1}\psi_1(\mathbf{x}), \ldots, \sqrt{\alpha_i}\psi_i(\mathbf{x}), \ldots)$$

where α_i and ψ_i are the eigenvalues and eigenfunctions of an integral operator whose kernel $K(\mathbf{x}, \mathbf{y})$ is a positive definite symmetric function. With this choice the scalar product in the feature space becomes particularly simple because:

$$\mathbf{z}^T(\mathbf{x})\mathbf{z}(\mathbf{y}) = \sum_{i=1}^{\infty} \alpha_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}) = K(\mathbf{x}, \mathbf{y})$$

where the last equality comes from the Mercer-Hilbert-Schmidt theorem for positive definite functions. The QP problem that has to be solved now is exactly the same as in eq. (1), with the exception that the matrix D has now elements $D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. As a result of this choice, the SVM classifier has the form: $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{\ell} \lambda_i y_i K(\mathbf{x}, \mathbf{x}_i) + b)$. In table (1) we list some choices of the kernel function proposed by Vapnik: notice how they lead to well known classifiers, whose decision surfaces are known to have good approximation properties.

It is worth noting that Haar wavelets are not the only choice for an appropriate dictionary. Smoother wavelets are very similar and may be more appropriate. Additional dictionaries of other basis functions (like Fourier basis) may be added to a basic dictionary of wavelets and may be especially desirable for certain tasks.

Our object detection framework at present does not use contextual information. However, as work by Lipson has shown, the qualitative template framework provides a convenient way to incorporate contextual knowledge. We are currently investigating whether the algorithms used by Lipson could be reformulated in terms of an overcomplete but fixed dictionary of basis functions such as wavelets.

3 The verification stage: Morphable models of object classes

The detection scheme described so far finds small regions in the image that are likely to contain the object sought. A verification stage is desirable to reject false alarms. For this goal we plan to use a newly developed model of classes of objects (such as faces) and to match this model to the image window of interest.

Our approach is based on modeling object classes as a linear combination of prototype images. The motivation for this model comes from the linear class concept [33]. They showed that linear transformations can be learned exactly from a small set of examples in the case of linear object classes. Since many object transformations can be approximated by linear transformations, this is a very useful result. As for practical applications which motivate this work, there are a number of important applications including man-machine interfaces, image analysis, video teleconferencing, image compression and object verification.

Cootes and Taylor [8, 7] proposed a similar model for object classes. They also use a linear combination of prototypes as an object model. Their work differs from ours in the algorithm they used to fit a model to an image, and in the use of a sparse set of control points as opposed to a full pixelwise correspondence matrix for their models. The work of Ullman and Basri [31] and Shashua [24] provided strong motivation for this work. They showed that any view of a single object can be represented as a linear combination of just three views of the object. Our work is concerned with extending this idea to object classes.

Kernel Function	Type of Classifier
$K(\mathbf{x},\mathbf{x}_i) = \exp(-\|\mathbf{x}-\mathbf{x}_i\|^2)$	Gaussian RBF
$K(\mathbf{x},\mathbf{x}_i) = (\mathbf{x}^T\mathbf{x}_i + 1)^d$	Polynomial, degree d
$K(\mathbf{x},\mathbf{x}_i) = \tanh(\mathbf{x}^T\mathbf{x}_i - \Theta)$	Multilayer Perceptron

Table 1: Some possible kernel functions and the type of decision surface they define

Our research focused on two main problems: modeling object classes and matching the model to a novel image. Our approach to the first problem is to model object classes as a linear combination of example images called prototypes. Linearly combining images means treating the images as vectors. In order to linearly combine these vectors, we must have pixelwise correspondence between the images. Otherwise, the linear combination would not form a vector space. So, in addition to the prototype images as input, we also require the pixelwise correspondences as input. The correspondences are represented as a flow fields from one prototype chosen as the reference prototype and each of the other prototypes. Given this, our model for object classes consists of two components, namely *shape* and *texture* vecors. The shape of a model image is a linear combination of prototype shapes (flow fields). Analogously, the texture of a model image is a linear combination of prototype textures (grey level images). The model thus contains two parameters per prototype: one for shape and one for texture. Adjusting these parameters allows one to create a large variety of model images by morphing the reference image according to the linear combination of shape and texture specified by the model parameters.

Our approach to the problem of matching a model to a novel image is the following. We first define an error measure between the input novel image and the current guess for the best fitting model image. This error is simply the L2 error between the pixels (grey level values) of the novel and model images. To obtain the model image, we must morph the reference image according to the model parameters. This produces an image against which we can compare the novel input image. Given this error function we find the best matching model image by optimizing the error with respect to the model parameters using a stochastic gradient descent algorithm.

The model and matching algorithm are described more fully in two papers in this volume (Jones and Poggio; Jones, Vetter and Poggio).

4 An application: an Intelligent Web Crawler

The problem of searching for information on the World-Wide-Web has become of critical importance with its enormous growth in the recent years. To address this problem new search engines and site indices have been developed (e.g.,Alta-Vista, Yahoo, Excite) and they have become an essential part of the Web. However, these search engine are based mainly on textual analysis (e.g., word statistics) of the site content or on categorical classification (e.g., business, sports, magazines, etc.). An important aspect of the Web is the inclusion of other media forms such as images and video. These media are not utilized for the task of information search. To take advantage of the visual content of the various Web sites, we have developed a search engine that is based on computer vision techniques which are applied to images collected on the web during the search. Such a search engine enables the user to search for sites that contain images of certain types (e.g. images of people or cars) or even to search for a particular person (e.g. Clinton).

4.1 System Architecture

The architecture of the search engine – as designed by Michael Oren and Jason Miller – is based on the Web-crawler scheme. The client (the user) sends to the server (the search engine) a URL address that is the starting point of the search. The web crawler visits the given address, locates all the images, and retrieves them to the server site. The server converts the images to a standard format and sends them to the image analysis module which provides information on the visual content of the image (e.g., does the image contain a frontal face?) Based on information that the server gets from the image analysis module it formats a result page and sends it back to the client. The server also identifies all the links to other sites (URL addresses) that follow from the current site and continues the search.

4.2 System Implementation

The web crawler engine was implemented using the Java programming language. The image analysis module was implemented in C/C++ since it is computationally intensive. Currently we are using a routine for face detection in a cluttered scene [28][17] that determines whether or not the image contains frontal faces and locates their positions.

The web crawler can be used not only as search engine but also as a pre-loader that download preselected information from the Web to the local disk. Pre-loading is a useful feature especially for users with low bandwidth Internet access and when viewing images on account of their large size. Unlike current commercial pre-loader software packages, which are limited to certain pre-defined sites, the vision-based web crawler can search new sites and download images based on their content.

We are working on adding a face recognition module which, together with the face detection routine, will enable the crawler to search for a specific person's image on the Web. We also plan to augment the web crawler with object detection and scene classification routines currently under development.

5 Cog: A Humanoid Robot

Although a primary focus of our collaborative work is in establishing a framework for designing search engines, we are also examining the utility, robustness and flexiblity of our components in a very different framework. This involves Cog – a ten degree of freedom upper torso humanoid robot with a fully mobile head/eye system, coupled to a with a six degree of freedom series-elastic actuated arm. Cog serves as an excellent platform on which to design, explore and evaluate visual modules, since the real-time performance capabilities of the robot provide a unique environment in which to test system designs. Many of the visual modules used in our visual search engines, especially low level feature extractors, focus of attention methods and tracking methods, have also been incorporated into Cog and tested in real time visual environments.

Further, Cog has served as a focal point for examining the trainability of different visual components. Examples include automatically training Cog to relate topographic maps from different modalities (e.g. oculomotor, visual tracking, auditory), to use such maps to orient to visual stimuli, to attend to distinctive visual stimuli, to learn ego-motion relationships, to locate and track faces, to learn hand-eye coordination and visually-guided pointing, and to coordinate stereo information with other perceptual cues.

We expect that as other components of our work in scene and object detection mature, they will also be incorporated into the growing suite of visual techniques available to our humanoid robot.

[1] D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of the International Conference of Computer Vision*, Cambridge, MA, June 1995.

[2] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272(5270):1905–1909, June 1996.

[3] I. Biederman. RBC: A theory of human image understanding. *Psych. Rev., 94,* 115-147, 1987.

[4] B.E. Boser, I.M. Guyon and V.N. Vapnik. A training algorithm for optimal margin classifier, *Proc. 5th ACM Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, 1992.

[5] R. Brunelli and T. Poggio. Template matching: Matched spatial filters and beyond. A.I. Memo 1549, MIT Artificial Intelligence Lab., 1995.

[6] C.J.C. Burges, Simplified Support Vector Decision Rules, *Int. Conf. Machine Learning*, pp. 71–77, 1996.

[7] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training Models of Shape from Sets of Examples, *British Machine Vision Conference*, pp. 9–18, 1992.

[8] T.F. Cootes and C.J. Taylor, Active Shape Models - 'Smart Snakes', *British Machine Vision Conference*, pp. 266–275, 1992.

[9] C. Cortes and V. Vapnik, Support Vector Networks, *Machine Learning* **20**:1–25, 1995.

[10] F. Girosi. Approximation error bounds that use VC-bounds. In *Proceedings of the International Conference on Neural Networks*, pages 295–302, Paris, October 1995.

[11] F. Girosi and N. Chan. Prior knowledge and the creation of "virtual" examples for RBF networks. In *Neural networks for signal processing, Proceedings of the 1995 IEEE-SP Workshop*, pages 201–210, New York, 1995. IEEE Signal Processing Society.

[12] M. Jones and T. Poggio. Model-based matching of line drawings by linear combination of prototypes. In *Proceedings of the International Conference on Computer Vision*, pages 531–536. IEEE, June 1995.

[13] M. Jones and T. Poggio. Model-based matching by linear combinations of prototypes. A.I. Memo 1583, MIT Artificial Intelligence Lab., 1996.

[14] P. Lipson, E. Grimson, P. Sinha, Configuration Based Scene Classification. *CVPR,* 1997

[15] P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8:819–842, 1996.

[16] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *(to appear in CVPR'97)*, 1997.

[17] Edgar Osuna, Robert Freund, and Federico Girosi. Support vector machines: Training and applications. *MIT CBCL-Memo*, May 1996.

[18] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *(submitted to NNSP'97)*, 1997.

[19] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *(to appear in CVPR'97)*, 1997.

[20] T. Poggio and D. Beymer. Learning networks for face analysis and synthesis. In Martin Bichsel, editor, *Proceedings of the International Workshop on Automatic Face and Gesture-Recognition*, pages 160–165. Zurich, Switzerland, 1995.

[21] T. Poggio and D. Beymer. Learning to see. *IEEE Spectrum*, pages 60–69, May 1996.

[22] R. Romano, D. Beymer, and T. Poggio. Face verification for real-time applications. In *Proceedings of the Image Understanding Workshop.* Morgan Kaufmann, San Francisco, 1996.

[23] H. Rowley, S. Baluja and T. Kanade. Human Face Detection in Visual Scenes. *School of Computer Science*, CMU-CS-95-158, Jul 1995.

[24] A. Shashua. Geometry and Photometry in 3D visual recognition, Ph.D. Thesis, MIT, 1992.

[25] P. Sinha. Perceiving and Recognizing Three-Dimensional Forms. Ph.D. Thesis, MIT Dept. of Electrical Engineering and Computer Science, 1995.

[26] P. Sinha and T. Poggio. Is human face recognition better characterized as head recognition? *Nature*, 384(6608), 1996. (Correspondence).

[27] P. Sinha and T. Poggio. Role of learning in three-dimensional form perception. *Nature*, 384:460–463, 1996.

[28] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *MIT AI Lab-Memo*, No. 1521, 1994.

[29] K.K. Sung and T. Poggio. Finding human faces with a gaussian mixture distribution-based face model. In *Proceedings of Second Asian Conference on Computer Vision*, Singapore, December 1995.

[30] K.K. Sung and T. Poggio. Finding human faces with a gaussian mixture distribution-based face model. In *Recent Progress in Computer Vision*, LNCS Series. Springer-Verlag, 1995.

[31] S. Ullman and R. Basri. Recognition by linear combinations of models, PAMI **13**:992–1006, 1991.

[32] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer:New York, 1995.

[33] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. A.I. Memo 1531, MIT Artificial Intelligence Lab., 1995.

Multidisciplinary Image Understanding Research at the University of Maryland

Azriel Rosenfeld
Center for Automation Research, University of Maryland
College Park, MD 20742-3275 (ar@cfar.umd.edu)

Abstract

This report summarizes research being conducted under the DoD MURI Program by the University of Maryland, the MIT Media Laboratory, and the University of Washington. Areas being investigated include motion modeling; human motion analysis; person tracking; face tracking and interpretation; target recognition; vehicle detection and tracking; self-tuning of IU algorithms; multi-scale discriminant analysis and recognition; robust estimation of optical flow; deformable contours; geometry of plane curves; model-based curve recognition; 3D curve reconstruction; and texture discrimination and description.

1 Introduction

The Computer Vision Laboratory of the Center for Automation Research (CfAR) at the University of Maryland is conducting research on many aspects of image understanding. This paper deals primarily with the research being conducted under Contract N00014-95-1-0521, entitled "Appearance-Based Vision for Complex Environments", which was funded under the Department of Defense's Multidisciplinary University Research Initiative (MURI) Program. The agent for this effort is the Office of Naval Research; the COTR is Dr. Harold L. Hawkins.

The MURI project at CfAR deals with video surveillance and tracking, with emphasis on complex scenes that contain humans or vehicles. Co-principal investigators on this project at CfAR are Rama Chellappa and Larry S. Davis. Also participating in the research are two subcontractors: the MIT Media Laboratory, Cambridge, MA (Principal investigators: Alex P. Pentland and Rosalind W. Picard); and the Department of Electrical Engineering at the University of Washington, Seattle, WA (Principal investigator: Robert M. Haralick). The following sections of this report briefly summarize the research being conducted on the project. Specific aspects of this research are described in greater detail in four separate papers in these Proceedings, as referenced below. The url for the project web page is http://www.cfar.umd.edu/cvl/MURI/.

2 Motion Modeling [Yacoob and Davis, 1997a, 1997b; Black *et al.,* 1997a; Black *et al.,* in preparation]

A model has been developed for computing image flow in image sequences containing a very wide range of instantaneous flows. This model integrates the spatio-temporal image derivatives from multiple temporal scales to provide both reliable and accurate instantaneous flow estimates. The integration employs robust regression and automatic scale weighting in a generalized brightness constancy framework. In addition to instantaneous flow estimation the model supports recovery of dense estimates of image acceleration and can be readily combined with parameterized flow and acceleration models. Its performance has been demonstrated on image sequences of typical human actions taken with a high frame-rate camera. Further details about the model can be found in a separate paper in these Proceedings.

As non-rigid and non-cohesive objects move or change state, their projected appearance onto the image plane of the camera changes. These changes in appearance due to deforming, articulating or eruptive/emergent events can be accounted for by two classes of image-based appearance events. The first class accounts for the deformations and articulations of the object in the image plane; we describe such *form* changes as "warps" of the object appearance in the image plane. The second class accounts for intensity changes resulting from occlusion, disocclusion, and changes in material properties of the object. We model these intensity variations by means of intensity templates and refer to them as *iconic* changes. The evolving appearance of a non-rigid and non-cohesive object is then modeled using a combination of form changes and iconic changes. We are developing a method for learning parameterized models of form changes and iconic changes, and an algorithm for recovering them from image sequences. These changes in image appearance can be used for recognition of various object deformations; we have illustrated this with examples of non-rigid and non-cohesive human mouth motion in natural speech.

3 Human Motion Analysis [Black *et al.,* 1997b; Horprasert *et al.,* 1997; Ju *et al.,* 1996; Yacoob *et al.,* to appear]

We have developed a representation in which we approximate the non-rigid motion of a person using a set of parameterized models of optical flow. While parameterized flow models (for example, affine flow) have been used for representing image motion in rigid scenes, Black and Yacoob observed that simple parameterized models can well approximate more complex motions if they are localized in space and time. Moreover, they showed that the motion of one body region (for example, the face region) could be used to stabilize that body part in a warped image sequence. This allowed the image motions of facial features (the eyes, mouth, and eyebrows) to be estimated relative to the motion of the face. Isolating the motions of these features from the motion of the face is critical for recognizing facial expressions using motion.

These parameterized motion models can be extended to model the articulated motion of the human limbs. Limb segments can be approximated by planes and the motion of these planes can be recovered using a simple eight-parameter optical flow model. Constraints can be added to the optical flow estimation problem to model the articulation of the limbs and the relative image motions of the limbs can be used for recognition. We have defined a "cardboard person model" in which a person's limbs are represented by a set of connected planar patches. The parameterized image motion of these patches is constrained to enforce articulated motion and is solved for directly using a robust estimation technique. The recovered motion parameters provide a rich and concise description of the activity that can be used for recognition. We are developing a method for performing view-based recognition of human activities from the optical flow parameters that extends previous methods to cope with the cyclical nature of human motion. We have illustrated our method with examples of tracking human legs over long image sequences.

We are also developing a method of estimating 3D head orientation in a monocular image sequence. The method employs recently developed image-based parameterized tracking methods for the face and face features to locate the area in which sub-pixel parameterized shape estimation of the eye's boundary is performed. This involves tracking of five points (four at the eye corners; the fifth is the tip of the nose). Our approach relies on the coarse structure of the face to compute orientation relative to the camera plane. It employs projective invariance of the cross-ratios of the eye corners and anthropometric statistics to estimate the head yaw, roll and pitch.

4 Real-Time Person Tracking (M.I.T. Media Laboratory) [Azarbayejani and Pentland, 1996; Pentland et al., 1997]

We have developed an estimation technique for recovering 3-D object shape and motion and multiple-camera geometry from 2-D features. The 3-D objects and 2-D features are both represented using moment-based physical models called blobs. Nonlinear optimization techniques are used for estimation; the Levenberg-Marquardt technique is used for static parameters and the extended Kalman filter is used for dynamic estimation.

This estimation method has been implemented as part of the M.I.T. Media Laboratories' Smart Spaces project. It is known as the STIVE system (Stereo Interactive Video Environment), and runs in real time using two SGI Indigo computers with no special hardware of any kind.

Experimental results show that the STIVE system can obtain good quantitative 3-D physical descriptions from coarse 2-D image observations of people. We have demonstrated that this method can be used to self-calibrate stereo cameras from watching people move, and subsequently to determine the locations, orientations, and shapes of parts of a person to an accuracy of 2 cm, 2 degrees, and a few percent, respectively (RMS errors).

Perhaps the most important performance evaluation, however, is that the STIVE system has run reliably for dozens of hours, with dozens of different subjects, in several different locations, and in real time (20-30 fps) using only standard workstations. The key to the robust, real-time performance is that the 2-D blob features on which the estimation relies can be reliably and efficiently extracted and matched in a bottom-up fashion.

We feel that use of this type of feature is a significant departure from the traditional notions of image features (e.g., points, lines) and image cues (e.g., motion fields, shading), and can lead to a basis for practical 3-D vision systems in application domains where traditional approaches have not had a great deal of success. Although the blob models provide only rigid motion and coarse shape information, they are fast and extremely reliable; thus further precision and higher levels of detail, if desired, can be safely bootstrapped from this level of representation, potentially leading to a powerful "coarse-to-fine" or "subsumption" approach to 3-D shape and motion analysis.

More recent work on real-time tracking and classification of human behavior is described in a separate paper in these Proceedings.

5 Real-Time Face Tracking and Interpretation (M.I.T. Media Laboratory) [Darrel *et al.,* 1996; Oliver *et al.,* 1996]

We have developed a real-time system for finding, tracking, and analyzing the human face and mouth. The LAFTER system (which stands for Lips And Face TrackER) has two outputs: (1) control of camera pan/tilt/zoom to maintain a well-centered image of appropriate resolution, and (2) recognition of mouth shapes using hidden Markov models (HMMs). The system runs on a single SGI Indigo computer, and produces 3-D estimates of head position that are surprisingly accurate. Classification accuracy for mouth shapes is consistently above 95% across all users.

Using a single SGI Indigo with a 200Mhz R4400 processor, the average frame rate for tracking is typically 25 Hz. When mouth detection and parameter extraction is added to the face tracking, the average frame rate is 14 Hz. The RMS error between the true 3-D location and the system's output was on the order of 1%.

Our approach to temporal interpretation of facial expressions uses HMMs to recognize different patterns of mouth movement. Using mouth shape as the input feature vector we trained five different HMMs for each of the following mouth configurations: neutral or default mouth position, extended/smile mouth, sad mouth, open mouth and extended+open mouth (such as in laughing).

Recognition results for eight different users making over 2000 expressions were typically about 98% on training and 96% on testing. The LAFTER system has also been tested on hundreds of users at different events, each with its own lighting and environmental conditions; the system performed well in almost every case, with most failures being attributable to dense beards or extreme skin color.

6 Target Recognition (University of Washington) [Liu and Haralick, 1997]

There are two difficulties with the classical use of templates for target recognition; one has to do with the spatial perturbation of the target in the image and the other has to do with the fact that the gray-scale perturbing noise is non-additive and non-Gaussian. Therefore when the template matching or matched filtering is done, the quantity computed is not related to the log probability of the data given the target. Our research has concentrated on understanding how to deal with the second effect, the effect of non-additive, non-Gaussian perturbation.

The idea behind the approach is to use in a template only those areas of the target which can effectively contribute to the detection and localization of the target. Using target areas which are spatially non-distinct and low-contrast not only does not contribute, but in effect can negatively affect detection accuracy and localization. The areas of the target that are least affected by non-additive Gaussian noise are those in which there are relatively high-contrast spatial structures. The high contrast will be the dominant effect relative to the non-Gaussian noise perturbation and will thereby permit good detection even with a technique which assumes an additive Gaussian perturbation. The spatial structure will permit good localization of the position of the detected target. Our work has concentrated on developing a methodology to automatically locate such areas on a target so that they can be assembled together as a template.

The technique relies on being able to analytically propagate the perturbation of the image data through to the covariance of the location estimated for the position of the target. To determine which additional target areas should be added to the template, we find the direction θ in which the detected target location has highest variance using the set of target areas currently in the template. Then we find the not yet used target area which when used for target location has smallest variance in the direction θ.

The theoretical basis of the covariance propagation method was published in (Haralick, 1996). This paper showed how to compute the covariance of any estimate obtained by either unconstrained or constrained optimization. It illustrated the application of the technique to a variety of computer vision problems. A discussion of this technique as it applies to target recognition is presented in a separate paper in these Proceedings.

7 Vehicle Detection and Tracking [Betke *et al.,* 1996]

A vision system has been developed that recognizes and tracks multiple vehicles from sequences of gray-scale images taken from a moving car in hard real time. Recognition is accomplished by combining the analysis of single image frames with the analysis of the motion information provided by multiple consecutive image frames. In single image frames, cars are recognized by matching deformable gray-scale templates, by detecting image features, such as corners, and by evaluating how these features relate to each other. Cars are also recognized by differencing consecutive image frames and by tracking motion parameters that are typical for cars.

The vision system utilizes the hard real-time operating system Maruti which guarantees that the timing constraints on the various vision processes are satisfied. The dynamic creation and termination of tracking processes optimizes the amount of computational resources spent and allows fast detection and tracking of multiple cars. Experimental results demonstrate robust, real-time recognition and tracking over thousands of image frames.

8 Design of Self-Tuning IU Algorithms [Shekhar *et al.,*1997]

Image Understanding (IU) systems used in challenging operational environments should provide both flexibility and convenience. Flexibility here means the ability to accommodate variations in the characteristics of the input data. Convenience means that an image analyst unfamiliar with the technical details of the IU system can obtain satisfactory results from it.

Flexibility is achieved by providing a number of tuning parameters, whose optimal setting usually ensures satisfactory performance. Such performance, however, is rarely achieved in practice since the strategies used by the IU system developer to find the optimal parameter settings are not conveniently available to the image analyst. The goal of our research is to achieve the conflicting goals of convenience and flexibility by providing a methodology for automatically achieving optimal parameter settings under operational conditions. The image analyst provides input in the form of qualitative evaluations of the results of IU processing, which are then interpreted in a rule-based framework to make the necessary adjustments to the appropriate algorithms. In this manner the IU system is given the capacity to tune itself for optimal performance.

In our previous work [Shekhar *et al.,* 1996], we discussed the knowledge-based semantic integration of IU algorithms using the OCAPI architecture. In this architecture, the reasoning of the IU specialist is formally represented using frames and production rules. Mechanisms are provided for program supervision tasks such as algorithm selection and tuning. We have extended this work to handle more complex program supervision strategies. We use the LAMA architecture [Vincent *et al.,* 1996] to implement our ideas. We are currently testing this approach on the vehicle detection algorithms developed at the University of Maryland [Chellappa *et al.*, 1996]. Details about this work can be found in a separate paper in these Proceedings.

9 Discriminant Analysis and Recognition [Etemad, 1996]

A successful pattern recognition scheme starts with efficient extraction of the most discriminant information elements from various, possibly imprecise, sources, followed by an intelligent combination of this information in a context-dependent framework of low complexity.

Conventional multiscale basis selection and feature extraction using criteria based on compression or approximation are not necessarily the best approaches for classification and segmentation purposes. Instead, a class separability based approach is preferable. We have developed methodologies for lower-dimensional adaptive multi-scale discriminant basis selection. Depending on the task, these methodologies are applied to local windows or to the whole pattern. Our tools in this analysis are derived from theories of wavelet packets and multi-scale local bases on the one hand, and from the statistical theory of discriminant cluster analysis on the other hand. The goal is to find efficient multi-scale representations that yield maximum between-class separations and minimum within-class scatters.

We have achieved improved classification reliability through context-dependent integration of soft decisions. We have investigated the effectiveness of soft decisions in representing the vagueness, uncertainty and imprecision of the classification sources. Based on the principle of least commitment in designing pattern recognition and consensus-theoretical concepts, we improve the reliability of our classification system through integration of soft decisions obtained from various observations and/or sources. The combination of decisions is based on the discrimination power of each source and its relevance to the current observation. Our approach uses ideas from consensus theory, fuzzy neural learning, and evidential reasoning.

Our methods of multi-scale local/global basis selection and context-dependent decision integration have been applied to several different domains, including texture and document image classification and segmentation, radar signature classification, and human face recognition. The results show that superior or highly competitive performance can be obtained using small feature sets and simple classifiers. The resulting systems are typically of low complexity and, since no iterative computations are involved, most of the calculations can be done in parallel. The proposed ideas can be extended in several directions and can be applied to many pattern recognition and segmentation tasks.

10 Optical Flow Estimation [Srinivasan and Chellappa, 1996]

Computation of optical flow has been formulated as nonlinear optimization of a cost function comprising a gradient constraint term and a field smoothness factor. Results obtained using these techniques are often erroneous, highly sensitive to numerical precision, and determined sparsely, and they carry with them all the pitfalls of nonlinear optimization. We have regularized the gradient constraint equation by modeling optical flow as a linear combination of an overlapped set of basis functions. We have developed a theory for estimating model parameters robustly and reliably. We prove that an extended least squares solution can be obtained that is unbiased and robust to small perturbations in the estimates of gradients and to mild deviations from the gradient constraint. The solution can be obtained by a numerically stable sparse matrix inversion, giving a reliable flow field estimate over the entire frame. Experimental results of our scheme are surprisingly accurate and consistent across a variety of images, in comparison with the standard optical flow algorithms. We believe that our flow field model offers higher accuracy and robustness than conventional optical flow techniques, and is better suited for image stabilization, mosaicking and video compression.

11 Deformable Contours [Gavrila, 1996]

We have developed a Hermite representation for deformable contour finding. This representation compares favorably in terms of versatility and controllability with other local contour representations that have been used previously for this purpose. The Hermite representation allows a compact representation of curved shapes, without the smoothing out of corners. It is also well suited for both interactive and tracking applications.

The Hermite representation has been used to formulate the contour finding problem as an optimization problem using a maximum a posteriori energy criterion. Optimization is performed by dynamic programming. Our approach to contour tracking decouples the effects of transformation and deformation, using a template matching strategy to robustly account for the transformation effect. We have demonstrated these ideas on a variety of images from different domains.

12 Differentialless Geometry [Latecki and Rosenfeld, 1996]

We have developed foundations for a theory of curves and surfaces that does not assume differentiability. Initial results have been obtained for plane curves, as described below; extensions to space curves and to surfaces are in progress. The concepts are also directly applicable to digital curves and surfaces.

Let S be a subset of the plane. A line l is called a *line of support* of S if S lies in one of the two closed halfplanes defined by l. We call S *supported* if there exists a line of support of S through every point of S. A set S is supported iff it is contained in the boundary of its closed convex hull; hence a closed, bounded, connected supported set must be an arc (or simple closed curve). A supported arc S has (finite, non-zero) one-sided derivatives at every point, and is differentiable at a point P iff the line of support of S at P is unique. (Note that the fact that S is an arc, and its differentiability properties, were not assumed; they are consequences of supportedness.) If S has a unique line of support at every non-endpoint, its curvature is defined at every non-endpoint and has constant sign, and its total absolute turn is at most 360^o.

An arc (not necessarily simple) is called *tame* if it is the concatenation of a finite set of supported arcs; for example, a polygonal arc is tame. A nonendpoint of a tame arc which is not interior to any supported subarc is called an *inflection*; if the arc is differentiable its curvature must change sign at such a point. A tame arc can have only finitely many inflections, and its total absolute turn must be finite.

13 Curve Recognition and Reconstruction[Weiss, 1995; Weiss, 1996]

It is well known there are no geometric invariants of a projection from 3D to 2D. However, given some modeling assumptions about the 3D object, such invariants can be found. The modeling assumptions should be sufficiently strong to enable us to find such invariants, but not stronger than necessary. We have developed such modeling assumptions for general 3D curves under affine projection. We show that if we know one of the two affine-invariant curvatures at each point of the curve, we can derive the other one from its image. We can also derive the point correspondence between the curve and the image.

There has been considerable work recently on the problem of reconstruction of 3D point sets from two images, taken by uncalibrated cameras. However, the point correspondence has to be given. When we deal with reconstruction of curves rather than points, while we need the correspondence between curves, this is an easier problem because curves are far fewer and more distinctive than points. We have derived a simple and general reconstruction method, based on an invariant coordinate system. We have applied it to non-coplanar conics and to combinations of a 3D conic with points. 3D cubics can also be handled. Unlike previous work, we do not need to know the epipolar geometry; we recover it from the images.

14 Texture Segmentation and Description [Ojala and Pietikäinen, 1996; Ojala, 1996]

A method has been developed for unsupervised texture segmentation utilizing a statistical test based on distributions of feature values to compare neighboring image regions. The distributions of simple local binary patterns combined with a complementary contrast measure are used as texture features and a log-likelihood test, the G test, is used for comparing feature distributions. A robust split-and-merge type algorithm is developed for coarse image segmentation. A pixelwise classification scheme with feature distributions of neighboring regions as texture models is then used to improve the localization at region boundaries. Our method does not require any prior knowledge about the number of textures or regions in the image. In experiments the method provides very good segmentations for various types of texture mosaics and natural scenes. The same set of parameter values is used in all the experiments. Generalizations of the method, e.g. to utilize other texture features, multiscale information, color features, and combinations of multiple features, are also possible.

A multichannel approach to texture description is realized by approximating joint occurrences of multiple features with marginal distributions, as 1-D histograms, and combining similarity scores for 1-D histograms into an aggregate similarity score. A stepwise feature selection algorithm is used to choose the best feature combination in a particular dimension. This approach gave excellent results in a classification problem which involved fifteen fine-grained textures from the Brodatz album. The statistical test used for measuring the similarity of sample and prototype histograms is not crucial in terms of overall performance; comparable classification results are obtained with a variety of tests. Further, the experimental results prove that choosing the proper quantization of the feature space is relatively easy.

15 Bibliography on Image Analysis and Computer Vision [Rosenfeld, 1997]

We have compiled a bibliography of nearly 2150 references related to computer vision and image analysis, arranged by subject matter, which appeared in 1996. The topics covered include computational techniques; feature detection and segmentation; image and scene analysis; two-dimensional shape; pattern; color and texture; matching and stereo; three-dimensional recovery and analysis; three-dimensional shape; and motion. A few references are also given on related topics, including geometry and graphics, compression and processing, sensors and optics, visual perception, neural networks, artificial intelligence and pattern recognition, as well as on applications.

References

A. Azarbayejani and A. Pentland, "Real-time Self-Calibrating Stereo Person Tracking using 3-D Shape Estimation from Blob Features". In ICPR'96, Vol. 3, pp. 627-632, Vienna, Austria, 1996.

M. Betke, E. Haritaoglu, and L. S. Davis, "Multiple Vehicle Detection and Tracking in Hard Real Time". CAR-TR-834, CS-TR-3667, June 1996. Presented at the IEEE Symposium on Intelligent Vehicles, Tokyo, Japan, 1996.

M. Black, Y. Yacoob, and D. Fleet, "Modeling Appearance Change in Image Sequences". To be presented at the Third International Workshop on Visual Form, 1997.

M. Black, Y. Yacoob, and S. Ju, "Recognizing Human Motion Using Parameterized Models of Optical Flow". To appear in *Motion-based Recognition*, edited by M. Shah, 1997.

M. Black, Y. Yacoob, A. Jepson, and D. Fleet, "Learning Parameterized Models of Image Motion". In preparation.

R. Chellappa, X. Zhang, P. Burlina, C.L. Lin, Q. Zheng, L.S. Davis, and A. Rosenfeld, "An Integrated System for Site-Model-Supported Monitoring of Transportation Activities in Aerial Images". In DARPA Image

Understanding Workshop, pp. 275–304, Palm Springs, CA, 1996.

T. Darrell, B. Moghaddam, and A. Pentland, "Active Face Tracking and Pose Estimation in an Interactive Room". In *IEEE* Conference on Computer Vision and Pattern Recognition, pp. 67-72, San Francisco, CA, 1996.

K. Etemad, "Multi-Scale Discriminant Analysis and Recognition of Signals and Images". CAR-TR-821, CS-TR-3629, April 1996.

D.M. Gavrila, "Hermite Deformable Contours". CAR-TR-817, CS-TR-3610, February 1996.

R.M. Haralick, "Propagating Covariance in Computer Vision". *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 10, pp. 561-572, 1996.

T. Horprasert, Y. Yacoob, and L. Davis, "Computing 3-D Head Orientation from Monocular Image Sequences". In International Conference on Face and Gesture Recognition, Killington, VT, pp. 242-247, 1996.

S. Ju, M. Black and Y. Yacoob, "Cardboard People: A Parameterized Model of Articulated Image Motion". In International Conference on Face and Gesture Recognition, Killington, VT, pp. 38-44, 1996.

L.J. Latecki and A. Rosenfeld, "Supportedness and Tameness: Differentialless Geometry of Plane Curves". CAR-TR-838, CS-TR-3686, September 1996.

G. Liu and R.M. Haralick "Automated Construction of Templates for Matching". In these Proceedings.

T. Ojala, "Multichannel Approach to Texture Description with Feature Distributions". CAR-TR-846, CS-TR-3731, December 1996.

T. Ojala and M. Pietikäinen, "Unsupervised Texture Segmentation Using Feature Distributions". CAR-TR-837, CS-TR-3685, September 1996.

N. Oliver, S. Pentland, F. Berard, and J. Coutaz, "LAFTER: Lips and Face Tracker". MIT Perceptual Computing Technical Report 396, 1996. http://www-white.media.mit.edu/vismod

A. Pentland, "Smart Rooms". *Scientific American*, Vol. 274, No. 4, pp. 68-76, April 1996.

A. Pentland et al., "Real-Time 3D Tracking and Classification of Human Behaviors". In these Proceedings.

A. Pentland, R. Picard, and P. Maes, "Smart Rooms, Desks, and Clothes: Toward Seamlessly Networked Living". *British Telecommunications Engineering*, Vol. 15, No. 3.

A. Rosenfeld, "Image Analysis and Computer Vision: 1996". CAR-TR-848, CS-TR-3733, January 1997.

C. Shekhar, P. Burlina, and S. Moisan, "Design of Self-Tuning IU Algorithms." In these Proceedings.

C. Shekhar, S. Kuttikkad, and R. Chellappa, "Knowledge-Based Integration of IU Algorithms". In DARPA Image Understanding Workshop, pp. 1528-1532, Palm Springs, CA, 1996.

S. Srinivasan and R. Chellappa, "Robust Modeling and Estimation of Optical Flow with Overlapped Basis Functions". CAR-TR-845, CS-TR-3721, December 1996.

R. Vincent, S. Moisan, and M. Thonnat, "A Library for Program Supervision Engines". Technical Report 3011, I.N.R.I.A., Sophia Antipolis, France, 1996.

I. Weiss, "Model-Based Recognition of 3D Curves from One View". CAR-TR-807, CS-TR-3581, December 1995.

I. Weiss, "3D Curve Reconstruction from Uncalibrated Cameras". CAR-TR-815, CS-TR-3605, January 1996.

Y. Yacoob and L.S. Davis, "Estimating Image Motion using Temporal Multi-scale Models of Flow and Acceleration". To appear in *Motion-based Recognition*, edited by M. Shah, 1997.

Y. Yacoob and L.S. Davis, "A Temporal Multi-scale Framework for Image Motion Computation". In these Proceedings.

Y. Yacoob, L. Davis, M. Black, D. Gavrila, T. Horprasert, and C. Morimoto, "Looking at People in Action—An Overview". To appear in Emphasis Computer Human Interface.

Advanced Visual Sensor Systems *

Terrance E. Boult
Rick Blum
Richard Wallace & Gary Zhang,
Ele. Eng. & Comp. Sci. Dept.
Lehigh Univ., Bethlehem, PA 18015
[tboult|rblum|rsw|gzhang]@eecs.lehigh.edu

Shree K. Nayar
Peter K. Allen
John. R. Kender
Department of Computer Science
Columbia Univ., NYC, NY 10027
[nayar|allen|jrk]@cs.columbia.edu

Abstract

In the battlefield of the future, advanced sensor systems will likely make the difference between the winners and the losers – both on the manufacturing and the military battlefields.

This report is on the Lehigh/Columbia MURI contract, which is a basic research project with its focus on sensors for manufacturing. Its basic research nature means that much of it can be applied for battlefield awareness or even medical applications. For example, Dr. Nayar's basic scientific work on imaging sensors with non-traditional optics led to the development of the omnidirectional sensor which has clear application in surveillance and tracking.

This report provides short summaries of our significant contributions, with citations to related papers. Length of presentation herein does not reflect level of effort nor our view of its significance – many of the most important areas have papers elsewhere in these proceedings.

1 Omnidirectional Video Cameras

Conventional video cameras are often limited in their fields of view; the image represents a small section of the scene that lies in front of the camera. Our interest is in the creation of video cameras that can "see" in all directions. Our omnidirectional cameras are based on the concept of catadioptric image formation; the use of mirrors in addition to lenses to form an image with an unusually large field of view, while maintaining a fixed viewpoint [Nayar-97]. We have developed several prototypes of omnidirectional cameras that use parabolic mirrors. In addition, we have solved the theoretical problem of deriving the entire class of catadioptric sensors that yield a single viewpoint [Nayar and Baker-97]. The fixed viewpoint in our cameras enables the user to create perspective (conventional) images of the scene from an omnidirectional image for any user-selected viewing direction and magnification [Nayar and Peri-97]. An interactive software system has been developed that allows a user to generate about a dozen perspective video ports from a single omnidirectional video stream [Peri and Nayar-97] using no more than a PC.

The implications of omnidirectional video for vision are numerous. We are presently developing a small number of our cameras for use by others in the DARPA IU community. The intended applications lie within the area of video surveillance, where omnidirectional cameras have the advantage that they do not have much of a blind spot. More information can be found in the PI-report on a new project centered on developing/applying omnidirectional video: system refinements, higher-level visual processing algorithms such as motion estimation, object tracking, object stabilization, and super-resolution algorithms that can be directly applied to omnidirectional video. Ultimately, the use of two or more omnidirectional cameras for full-view 3-D reconstruction of scenes will be pursued.

2 Rational Filters for Passive Depth from Defocus

Last IUW we demonstrated a significant breakthrough in range image acquisition – a defocus-based system that produced 512x512 depth maps at 30fps. The active illumination limits its use outdoors.

Our past successes have motivated us to develop a general purpose sensor that captures, simultaneously, two images of the scene taken at different focus settings. This bifocal sensor is comprised of a telecentric lens [Watanabe and Nayar-95], a beam-splitting

*This work supported in part by ONR/DARPA MURI program ONR N00014-95-1-0601. Several other agencies and companies have also supported aspects of this research. For more infomation visit:
http://www.eecs.lehigh.edu/VAST
http://www.cs.columbia.edu/CAVE
http://www.cs.columbia.edu/robotics

prism, and two CCD image sensors. The telecentric lens ensures that the magnification of the sensor is invariant to defocus. The two CCD sensors are positioned to provide slightly different effective focal lengths. As a result, two images of the scene are sensed that correspond to different focus settings.

A central component in this passive depth from defocus system is a new class of broadband *rational operators* that, when used together, provide invariance to scene texture and produce accurate and dense depth maps [Watanabe and Watanabe-96, Watanabe and Watanabe-98]. Since the operators are broadband, a very small number of them are sufficient for depth estimation of scenes with complex textural properties. In addition, a *depth confidence* measure has been derived that can be computed from the outputs of the operators. This confidence measure permits further refinement of computed depth maps. Experiments were conducted on both synthetic and real scenes to evaluate the performance of the proposed operators. The depth detection gain error is less than 1%, irrespective of texture frequency. Depth accuracy is found to be 0.5 ~ 1.0% of the distance of the object from the imaging optics.

We are presently developing algorithms that use the bifocal sensor for real-time depth map computation, real-time depth-based image segmentation, and real-time depth-based image compression.

2.1 CAD Model Acquisition from Multiple Range Images

Some aspects of our project are more closely tied to the manufacturing area; CAD model acquisition is one of them. Direct recovery of an accurate CAD model from an object would also be beneficial for battlefield simulation development and for part replacement. Furthermore the basic issues of where-to-sense and how to combine the polygonal data in a consistent manner transcend the CAD arena.

We have developed a new method for automatically constructing a CAD model of an unknown object from range images. The method is an incremental one that interleaves a sensing operation that acquires and merges information into the model with a planning phase to determine the next sensor position or "view" [Reed and Allen-97a, Reed and Allen-97b, Reed *et al.*-97b, Reed *et al.*-97a]. This is accomplished by integrating a system for 3-D model acquisition with a sensor planner. The model acquisition system provides facilities for range image acquisition, solid model construction and model merging: both mesh surface and solid representations are used to build a model of the range data from each view, which is then merged with the model built from previous sensing operations. The planning system utilizes the resulting incomplete model to plan the next sensing operation by finding a sensor viewpoint that will improve the fidelity of the model. Experimental results can be found in these proceedings [Reed *et al.*-97b] for complex parts that include polygonal faces, curved surfaces, and large self-occlusions.

In the first phase of our method, we acquire a range image, model it as a solid, and merge it with any previously acquired information. This phase motivates the generation of a topologically correct solid model at each stage of the modeling process, which allows the use of well-defined geometric algorithms to perform the merging task and additionally supports the view planning process. The second phase plans the next sensor orientation so that each additional sensing operation recovers object surface that has not yet been modeled. Using this planning component makes it possible to reduce the number of sensing operations to recover a model: systems without planning typically utilize as many as 70 range images, with significant overlap between them. This concept of reducing the number of scans is important for tasks such as 3-D FAX where the sensing process may add considerable time. The result is a 3-D CAD model of the object.

For each range scan, a mesh surface is formed and "swept" to create a solid volume model of both the imaged object surfaces and the occluded volume. This is done by applying an extrusion operator to each triangular mesh element, sweeping it along the vector of the rangefinder's sensing axis, until it comes in contact with a far bounding plane. The result is a 5-sided triangular prism. A regularized union operation is applied to the set of prisms which produces a polyhedral solid consisting of three sets of surfaces: a mesh-like surface from the acquired range data, a number of lateral faces equal to the number of vertices on the boundary of the mesh derived from the sweeping operation, and a bounding surface that caps one end. Each surface is tagged as "seen" or "occlusion" for the sensor planning phase that follows.

Each successive sensing operation will result in new information that must be merged with the current composite model. The merging process itself starts by initializing the composite model to be the entire bounded space of our modeling system. The information determined by a newly acquired model from a single viewpoint is incorporated into the composite model by performing a regularized set intersection operation between the two. The intersection operation must be able to correctly propagate the surface-type tags from surfaces in the models through to the composite model.

Critical to any incremental method is planning the next view of the object to reduce uncertainty. What is desired is a method that takes known self-occlusions into account, and yet does not need to discretize the sensing positions and compute an image for each of them. Our method is able to select a specific target to be imaged which will reduce the uncertainty and is also able to avoid self-occlusion problems. Once an unoccluded sensor position for the specified surface has been determined, it may then be scanned, modeled, and integrated with the composite model. Our method is target-driven and performed in continuous space. As the incremental modeling process proceeds, regions that require additional sensing can be guaranteed of having an occlusion free view from the sensor if one exists. Other viewing constraints may also be included in the sensor planning such as sensor field of view, resolution, and standoff distance of the sensor.

2.2 Dynamic Sensor Planning

We have extended our Machine Vision Planning (MVP) system [Tarabanis *et al.*-94, Tarabanis *et al.*-95a, Tarabanis *et al.*-95b] which automatically computes viewpoints for monitoring objects and features in a robot work-cell, to function in an environment in which objects are moving [Abrams-97]. There are three key elements of this research. The first component is the computation of the volumes swept out by moving objects over a time interval. This permits the determination of occluded regions during that time interval. We have developed a new algorithm for swept volume computations, handling arbitrary polyhedral objects moving through arbitrary trajectories. We have also developed a new algorithm for the computation of the camera positions, orientations, and optical settings to be used during these time intervals for robotic inspection and monitoring tasks. These algorithms may have further application in automated surveillance applications where camera positions have to be chosen a priori. The conflicting nature of the viewing constraints (e.g. resolution and field-of-view tend to want different solutions) has led us to develop a new method for viewpoint computation. This method [Abrams *et al.*-96] effectively decomposes the constraints of focus, resolution, field-of-view, and visibility into satisfiable subsets that can then be efficiently searched to reach an overall solution. A five-degree-of-freedom Cartesian robot carrying a CCD camera in a hand/eye configuration and surrounding the work-cell of a Puma 560 robot has been constructed for performing sensor planning experiments. The results of these experiments, demonstrating the use of this system in a robot work-cell, are described in detail in [Abrams-97].

We briefly describe one experiment here. In circuit board assembly inspection, certain locations on the IC board need to be visually monitored at all times, even though possibly occluded by a moving robot in the workcell. Our system models the workcell and any motion in it, creating swept 3-D volumes for any object in the workcell whose motion is known. Using these swept objects, which can be thought of as a temporal occlusion objects, we can compute the visibility volumes for each location that is to be monitored. Optical constraints are then solved for and merged to find a correct viewpoint if one exists. If a single view is not sufficient, the volumes are decomposed to find intermediate viewpoints that suffice.

3 Parametric Feature Detection

In [Baker *et al.*-98a, Baker *et al.*-98b, Nayar *et al.*-96b] we have developed an algorithm to automatically construct a feature detector for an arbitrary parametric feature. Our work has resulted in a general framework for feature detection; it permits a user to input a parametric feature model and provides a feature detector that is optimal in the correlation sense. Besides the unparalleled generality of the technique, the other major contribution is the incorporation of realistic models of optical and sensing effects, which we argue to be vital in order to obtain a high level of feature detection robustness. In the algorithm, each feature is represented as a densely sampled parametric manifold in a low dimensional subspace of a Hilbert space. Detection is then performed by projecting the brightness distribution around each image pixel into the subspace. If the projection lies sufficiently close to the feature manifold, the feature is detected and the location of the closest point on the manifold yields the feature parameters. To find the closest manifold point sufficiently quickly, we employ parameter reduction by normalization, dimension reduction [Nayar *et al.*-96b], pattern rejection [Baker and Nayar-96], and heuristic search [Baker *et al.*-98b].

We have applied the algorithm to construct detectors for 5 parametric features, namely, step edge, roof edge, line, corner, and circular disc. Using these 5 detectors, we have conducted detailed experiments to demonstrate the efficacy of the technique and in particular have found it to perform comparably to the edge detectors of Canny [Canny-86] and Nalwa-Binford [Nalwa and Binford-86]. As future work, we are interested in the analysis of weighting functions that optimize the performance of feature detection. This will involve a theoretical analysis of the formulation of feature detection measures as well as the numerical optimization of weighting functions.

4 Deformable models

Deformable models are used for tracking and object recognition. We are investigating inductive learning of deformable models. Such models find the shape that best describes the boundary of some object in an image. That is, given an image, some objective ("energy") function is optimized over a family of parametrized shapes.

In cluttered environments, simple gradient strength at the shape boundary does not distinguish the desired object from others. Other image and shape qualities must guide segmentation, and training on domain data determines how. Prior work with training has used particular features besides gradient strength, but none has examined training in a general setting. Our work establishes a methodology for selecting image features and objective functions learned from them, and for measuring their effectiveness in order to find the best way to optimize a parameterized shape within cluttered imagery in a domain.

In our work, the input to the training is a large set of presegmented images in which the sought shape ("ground truth") is known. A probability model of the chosen image features describes a family of objective functions. Given the model, the training data determines the objective function. The image features are quantities that relate the shape to the image, and include the usual intensity and gradient measures, but they are potentially unlimited and may even be non-local. For example, one such non-local feature could measure the distance or orientation to a remote but reliable landmark.

We then evaluate the quality of the features-plus-function combination. This is done statistically; we generate random perturbations of the ground truth shape in presegmented images, and then examine the relationship between the amount of perturbation and the perturbed shape's objective function value. Initial results are on medical imagery but they will also be applied in the manufacturing domain.

5 Warping-based fusion & Super-Resolution

Given an image sequence from one or more sensors, it seems natural to want to fuse the data to produce new and better estimates. Image fusion almost always requires warping to align the data before the fusion step. One of our ongoing projects is improved algorithms for this warping and various algorithms for image fusion. The techniques has been applied to warp 4 camera views to compute polarization images[Zhou-96], and for developing super-resolution images [Chiang and Boult-97b, Chiang and Boult-97a, Chiang and Boult-96a].

There are, of course, some fundamental limits on what this combination can do. If the images were noise-free, focused and Nyquist sampled, then multiple images from a single viewpoint would add nothing. However, things move, images are blurred and with the noise, and aliasing present in images, deblurring is unstable. If time is not a concern, then existing techniques can address these problems formulating fusion as millions of coupled equations. For industrial or battlefield use, the fusion needs to be fast - a four fold increase in resolution that takes 4 hours is not likely to get used. Other researchers, e.g. [Huang and Tsai-84, Irani and Peleg-91, Irani and Peleg-93, Bascle *et al.*-96], have addressed this problem making various accuracy/time tradeoffs.

We have developed a technique that can warp/fuse images in seconds [Chiang and Boult-96a] with quality superior to the best previous work. Unlike previous approaches our algorithm is direct, not iterative. The algorithm performs matching, warping, and fusion followed by an explicit (optional) deblurring/sharpening step. We showed that image warping techniques can have a strong impact on the overall quality. By coupling a degradation model of the imaging system directly into our integrating resampler[Chiang and Boult-96b], we can better approximate the warping characteristics of real sensors, which in turn improves the quality of super-resolution images. Quantitative evaluation is underway, passing the resulting super-resolution images to recognition systems and comparing recognition performance. Initial results support our claims of superiority.

Until now, super-resolution algorithms required the images be taken under the same illumination. This is acceptable for very short video-sequences from a single viewpoint in a controlled environment. For military situations, there was a desire to obtain super-resolution images taken over significant time spans and different viewpoints. When intensities change, simple fusion cannot estimate them.

We recently demonstrated a new approach to super-resolution which uses edge models, local blur estimates and intensity reconstruction from these models to circumvent the problems of lighting variations [Chiang and Boult-97a, Chiang and Boult-97b]. We fuse estimates of the sub-pixel edge and blur models and then reconstruct the desired super-resolution intensity image. A remaining significant problem is the required sub-pixel matching which is more difficult with changing illumination, and even more so with viewpoint changes. We are currently doing a quantitative evaluation of both of our super-resolution algorithms in domains where the matching is not a significant problem. This edge/blur based super-resolution approach represents a fundamental advance for warping-based image fusion.

6 Region plus Wavelet-based Fusion

Not all fusion can be accomplished at the "pixel" level. We have studied a new fusion concept [Zhong and Blum-97] which combines the traditional pixel-level fusion with feature-level fusion. Under this concept, we developed a new region-based fusion algorithm in which the wavelet transform, edge detection and image segmentation are all combined. Experiments [Zhong and Blum-97] show that this algorithm works well in many situations. The same approach works equally well for fusion of different sensors, the same sensor with different parameters and scenes with different lighting. One demonstration, see [Zhong and Blum-97] in these proceedings, uses a pair of visual and 100GHz radiometric (i.e. mm-wave radar) images. The visual image shows location and appearance of the people while the radiometric image shows the existence of a gun. From the fused image, one can clearly and promptly see who, if anyone, concealed a gun beneath their clothes [Currie *et al.*-95, Currie *et al.*-96, Zhong and Blum-97].

7 Fusing Decisions

Sensory data will ultimately be used to make some decision on the state of some device, part, or process. Given that mechanisms exist for making individual decisions based on the measurements of each of the individual sensors acting alone, methods for combining the individual decisions are clearly of interest. These decision fusion methods, [Blum *et al.*-97], should account for, among other things, the differences in the reliabilities of the decisions made by each of the different sensors. In our research the observations may be statistically *dependent* across sensors which is more realistic than past "independent" work. In [Blum-95] we proposed and analyzed an adaptive scheme for learning the appropriate decision combining rule and prove our combining rules will convergence to an optimum solution (minimum probability of error). Numerical tests indicate that convergence typically occurs quickly.

There are clear advantages to tuning the individual decision mechanisms at the same time that the decision combining scheme is being adapted. We have developed an approach [Deans and Blum-96, Deans-96] which enables the application of existing adaptive filtering and processing schemes and tests of this approach are promising. Initially our approach was limited to a specific class of observation models which include the difficult case where the signals of interest are weak and the observations are corrupted by either additive Gaussian or non-Gaussian noise. More recently we have formulated an approach which can be applied to a more general set of problems [Shen-97].

We have begun developing a theory for the optimum decision rules for the individual decision makers in decision combining schemes with dependent observations such a theory. In our recent research [Vikalo-97, Vikalo and Blum-97b, Vikalo and Blum-97a] the noise was modeled as a mixture of Gaussian distributions, a general and practical model for impulsive noise. A criterion of Bayes risk is adopted for cases with fixed fusion rules. The optimum sensor tests are shown to be different from the best isolated sensor tests in several cases. Further, a methodology for predicting the form of the optimum sensor tests has been developed, see [Vikalo and Blum-97b].

8 Closest Point Search in High Dimensions

Closest point search is an important component of many recognition techniques in computational vision. For instance, in appearance based recognition [Murase and Nayar-95a], positioning, tracking, and inspection [Nayar *et al.*-96a], the point in eigenspace closest to a novel input point identifies an appearance which is most similar to the appearance that the novel point represents. Existing closest point algorithms, such as indexing [Califano and Mohan-91], k-d tree, and R-tree [Guttman-84] are not efficient in high dimensional spaces. As a result they perform poorly when employed for appearance recognition, since eigenspaces typically have more than 15 dimensions.

We have developed [Nene and Nayar-96] a new algorithm which substantially improves high dimensional performance. In our technique, rather than searching for the closest point, we search for a closest point within distance ϵ from the novel point. This is accomplished by partitioning the space with pairs of hyperplanes, one pair for each dimension and each hyperplane placed at distance ϵ from the novel point. The intersection of these hyperplanes gives a hypercube, the points within which can be found efficiently with the aid of a precomputed data structure. An exhaustive search is then performed on these small number of hypercube points to find the closest point. In [Nene and Nayar-96], we examined the complexity of this algorithm and show that, for commonly occurring distributions, the complexity is roughly $O(nd\epsilon)$, where n is the number of points and d the number of dimensions. Thus, we trade exponential behavior in the number of dimensions to linear behavior in number of points. Extensive benchmarks demonstrate that our algorithm outperforms competitive techniques by a large margin [Nene and Nayar-85].

Closest point search is a generic problem and the newly developed technique could also be applied in image databases, pattern analysis, and various types of simulation.

9 Visual Gestural Interfaces

Last IUW we reported initial work on visual gestural interfaces. Since then the work has undergone evaluation and extensions. This work takes as input images of an uninstrumented hand against a natural background, uses standard image processing hardware to segment the hand via color matching, tracks the hand, and at critical junctures, analyzes the pose via a neural net [Kjeldsen and Kender-96b, Kjeldsen and Kender-96a]. The gestural class of the pose, and the positional information of the tracking, can be used to drive a standard menu-based user interface: for example, selecting a menu, pulling it down, and "clicking" on an item by gestures, or moving and resizing the windows themselves.

The first study establishes the non-linearities involved in displaying the tracked cursor position given the visual input. We show that the smoothing constraints of the tracking are critically dependent on the context in which hand motion occurs. Modeling the cursor as a physical object with "mass", position, and velocity meets some but not all of these constraints. We improved the system by detecting various contexts and dynamically adjusting the smoothing parameters depending on apparent user intention. The "force" function, which transmits visual location to cursor position via a sigmoidally varying "spring" constant, depends on current and prior positions and velocities.

In the second study, we compared the usability of the visual interface with that of standard pointing and clicking in several ways. First, we studied the accuracy and repeatability of visual tracking in an alternating target task. The new smoothing algorithm effectively damps out the majority of the jitter that is present in the raw hand position data, but nevertheless tracks fast movements very well. Next, we evaluated object selection performance directly, by measuring the length of time needed to select a target by pointing at it, and compared this time to that using a mouse. Selection time was measured from the moment the space key was pressed until the cursor had been inside the target continuously for 0.5 seconds. The mean selection time for free-hand pointing was 1.91 seconds; the mean selection time using the mouse was 1.57. These times include the 0.5 seconds within the target. However, free-hand pointing time drops rapidly with increasing target size, leveling out at around 1.2 seconds for larger objects; selection time with a mouse drops only to 1.3 seconds.

Lastly, we developed a predictive model for the accuracy of our free-hand pointing according to Fitts' Law. The model accurately captured both hand data and mouse data and allows us to predict system performance as a function of tracking rate and tracking accuracy. The model shows that random jitter in the cursor position and the lag caused by the slow tracking rate are sufficient to cause the long selection times for small objects. The model indicated that accuracy was far more critical than speed; with very little noise and at a tracking speed attainable with off-the-shelf hardware in a few years, free-hand pointing can be expected to be approximately the same as for a mouse, slightly better for large objects, and slightly worse for small ones. Under ideal conditions (i.e. no tracking lag at all), gesture has the potential to be significantly faster than using a mouse for objects of all sizes.

9.1 Visual Control of Grasping

While great strides have been made in robotic hand design and a number of working dextrous robotic hands built, the reality is that the sensory information required for dextrous manipulation lags the mechanical capability of the hands. Accurate and high bandwidth force and position information for a multiple finger hand is still difficult to acquire robustly. Vision can be an effective sensing modality for grasping tasks and can serve as an external sensor to provide control information for devices that lack internal sensing or that would require extensive re-engineering to provide contact and force sensing. Using a vision system, a simple uninstrumented gripper/hand can become a precision device capable of position and possibly even force control. When vision is coupled with any existing internal hand sensing, or external tactile sensors, it can provide a rich set of complementary information to confirm and quantify internal sensory data, as well as monitoring a task's progress [Yoshimi and Allen-97].

We have implemented a set of real-time vision modules that can be used to track and monitor the hand as it performs a task. The vision system is used to track the links of each finger of the hand as well as monitor contact and grasp points on objects in the workspace. We have also added tactile sensors to augment the capability of our robotic hand [Allen *et al.*-97, Allen *et al.*-96a, Allen *et al.*-96b]. The tactile sensors cover the links of the hand as well as the palmar surface. The tactile sensor system can be used to localize contacts on the surfaces of the hand, as well as determine contact forces. The tactile pads use a capacitive tactile sensor and the electronics package is mounted on the robot wrist with wiring to each pad on the fingers and palm. The tactile sensor geometry on each finger link is a 4x8 grid with each capacitive cell approximately 3 mm by 3 mm and 1 mm spacing between tactile elements (tactels), and the sensor can bend to the curve of the fingertip. The sensor is covered with a compliant elastomer that allows force distributions. The robotic hand we are using is the Barrett Hand, which is a three-fingered, four DOF hand with limited sensing capability. It has a limited amount of internal strain gauge force sensing capability built into it, and the tactile and vision systems can be used to accurately quantify contact forces in conjunction with the strain gauge system.

A number of experiments have been performed with this system to characterize the ability of vision and force/contact sensors to support grasping tasks. They include integration of real-time visual trackers in conjunction with internal strain gauge sensing to correctly localize and compute finger forces, determination of contact points on the inner and outer links of a finger through tactile and visual sensing, and determination of vertical displacement by tactile sensing for a grasping task. In these experiments, the vision system reported contact points that were within 2 mm of the actual contact points.

10 Remote control of sensors/actuators

In support of the Laptop Vision System we have developed a set of software drivers allowing remote control of actuators. The high-level interface is web oriented and the low level drivers provide sufficient intelligence so as to handle reasonable delays in the command stream.

The first layer was development of some much-needed software to generate PWM (pulse-width-modulated) signals directly on the PC parallel port under control of commands received on the PC serial port. Both "locked anti-phase" and the more difficult "sign-magnitude" modes of PWM generation are supported for maximum flexibility in driving different kinds of H-bridges. First order prediction of the duty cycle with respect to time is explicitly supported so that busy hosts can afford to be a little late in sending commands on the serial line, yet still produce smooth actuator trajectories. We also developed a server which presents, to any number of clients, a high-level TCP/IP control interface. Clients include two GUI interfaces that allow one to manipulate the direct-drive motors with on-screen slider bars, with radio-button selections and, in the case of the two-axis motors (such as the SPM), with direct 2D position maps.

A parallel development has been the TCP/IP video server, intended principally to allow clients to capture and manipulate logmap images in real time, but designed in a modular fashion that will easily accommodate many kinds of filter. The server handles the sequencing and synchronizing of image capturing and the filtering pipelines.

As a demonstration of the servers, we developed a WWW-program that allows users on the WWW (http://www-robotics.eecs.lehigh.edu:8009/imp]) to point our video camera mounted on a spherical pointing motor. The default filter gives a logmap view, and images can be passed through any number of subsequent filters. This graphical client uses calibration information to map positions on the current image to the PWM values required to point to those positions. Unlike past web-cams, this interfaces adjusts the view to what the user specifies, not just in some user specified direction. A third client is Richard Wallace's "Alice" program whose Web clients can use a natural-language interface to move the camera.

11 Recovery of Textureless Scenes

It is a widely held belief in computer vision that textureless surfaces cannot be recovered using passive measurement techniques. Well known methods such as stereo and structure from motion need correspondences between points in order to enable the recovery of the scene. This is impossible in the absence of scene texture. Monocular techniques such as depth from focus and depth from defocus also require that the scene be strongly textured. A partial solution to the problem is to use shape from shading, but it is limited by strong assumptions in order to make the problem tractable.

All the methods described above try to recover the structure of the scene by analyzing a set two dimensional images. We formulate a novel approach to the problem by viewing the process of image formation as a fully three-dimensional mapping [Sundaram and Nayar-97]. We model the process by which the lens encodes the three dimensional volume behind the lens (which we define to be the Monocular Visual Space (MVS)) with the structural information of the scene in front of the lens. By analyzing the properties of the MVS, we have determined necessary and sufficient conditions to recover from the MVS the structural information of the scene. These conditions led to a simple procedure to recover textureless scenes. We have demonstrated experimentally the recovery of three classes of textureless surfaces: planes, lines and paraboloids. The conditions and the methods that we have proposed for scene recovery are general in nature and are applicable to all scenes and are not limited to textureless scenes. Textureless scenes merely represent the worst case scenario for recovery.

12 Reflectance and Texture of Real-World Surfaces

The appearance of real-world textured surfaces depends on view, illumination and the scale at which the texture is observed. At coarse scale, appearance is characterized by the BRDF (bidirectional reflectance distribution function). At fine scale, appearance can be characterized by the BTF (bidirectional texture function). The problem of characterizing surface appearance in terms of the BRDF and BTF has been addressed in [Dana *et al.*-97] and has resulted in three publicly available databases: a BTF measurement database with texture images from over 60 different samples, each observed with over 200 combinations of viewing and illumination directions, a BRDF measurement database with reflectance measurements from the same samples and a BRDF model parameter database with parameters obtained by fitting the measured data to two recent BRDF models. Specifically, the measurements are fit to two existing analytical representations: the Oren-Nayar model [Oren and Nayar-95, Nayar and Oren-95] for surfaces with isotropic roughness and the Koenderink et al. decomposition [Koenderink *et al.*-96] for both anisotropic and isotropic surfaces. These databases are publicly available at `www.cs.columbia.edu/CAVE/curet`.

Exactly how well the BRDFs of real-world surfaces fit existing models has remained unknown as each model is typically verified using a small number (2 to 6) of surfaces. Our BRDF measurement database allows us to evaluate the performance of existing models as well as future models. The BRDF parameter database is a concise representation of the measurements that can be directly used for both image analysis and image synthesis. The BTF database can be used for development of algorithms for 3D texture, i.e. image texture due to surface roughness. The changing appearance of 3D texture with view and illumination cannot be studied using existing texture databases which have few images (often a single image) for each sample. This work represents the first comprehensive investigation of a large variety of real-world surfaces. Our future work is geared towards the recognition and synthesis of real-world textures and will rely heavily on the use of the three databases we have created.

13 Ordinal Measures for Visual Correspondence

Most traditional approaches to matching visual information rely on correlation based measures. In contrast, ordinal measures are based on relative ordering of intensity values in a image region called rank permutation [Bhat and Nayar-96, Zabih and Woodfill-94]. By using distances between permutations [Bhat and Nayar-96, Gideon and Hollister-87], an entire class of ordinal measures can be arrived at. In [Bhat and Nayar-96], we showed the utility of these measures for pixel-by-pixel stereo correspondence where the chief issues were robustness in the presence of depth discontinuities, occlusion, and noise. We also discussed methods for efficient computation of the measures which is important for practical application.

In [Bhat *et al.*-97], we presented a method for motion estimation using ordinal measures. While popular measures like the sum-of-squared-difference (SSD) and normalized correlation (NCC) rely on linearity between corresponding intensity values, ordinal measures only require them to be monontonically related so that rank permutations between corresponding regions are preserved. This property turns out to be very useful for motion estimation in, for instance, tagged magnetic resonance images. We studied the imaging equations involved in two methods of tagging and observed temporal monotonicity in intensity under certain conditions though the tags themselves fade. We compared our method to SSD and NCC in a rotating ring phantom image sequence. We have investigated computational issues and presented experiments that demonstrate the use of ordinal measures for motion estimation. Our future work will be focused on the use of ordinal measures for image retrieval from large databases. We expect the measures to perform more robustly than existing ones based on correlation.

14 IUE development

We have continued to take an active role in the IUE development, working on system internals, education and library development.

One of our IUE contributions includes modification to the code-generation system and other system internals to help dramatically reduce the size of the compiled libraries – while functionality and classes have been continually added to the system, the libraries' size has been reduced by a factor of about 4.

We have been active in the education of the community about thc IUE. Lehigh hosted, and Dr. Boult led, IUE summer camps in both 1995 (6 weeks) and 1996 (2 weeks), training approximately 20 researchers in IUE development. We also helped teach an IUE summer training camp at INRIA for another 15 researchers, and organized 3 workshops/one day tutorials.

The final aspect of our IUE work concentrates on the porting of SLAM into the IUE. SLAM (Software library for Appearance Matching) was presented at last year's IUW and has been demonstrated on object recognition [Murase and Nayar-95b, Nayar *et al.*-95], visual tracking [Nayar *et al.*-94] and other applications. SLAM has been successfully tested on a database with over 100 3D objects many of which are difficult using pure geometric approaches. We felt this was an important package to bring into the IUE. As SLAM was not developed with IUE integration in mind, we felt it also would present a realistic integration challenge.

Last fall we demonstrated an initial integration, which was done by adding data conversion between IUE data and the SLAM functions and providing "wrappers". However, this decreased performance and increased the memory usage. A systematic approach was adopted to truly integrate SLAM into the IUE, *without* altering the stand-alone SLAM code. This highlighted a weak spot of the IUE, it did not support raw memory sharing between vector, matrix and image classes, or with other similar external data blocks. We developed a general solution using a reference counted smart pointer block-data class for the IUE [Zhang and Boult-97]. This class allows data between difference classes to be shared, and should make it easier for any other applications that want to "share" their data with the IUE.

Full evaluation of the ported SLAM is still in progress to quantify the performance characteristics, e.g. tolerance to small features, the selection of parameters and the prediction of eigen-vector selection on the classification results. We will also be exploring re-implementation of important components using all integer mathematics, and development of a user interface for the IUE version of SLAM.

15 Low Cost Laptop Vision

One of our project goals is to integrate Common-Off-The-Shelf into a portable and low cost vision and image processing system. The Laptop Vision System (LVS) can process disparate sources of sensory information, e.g. intensity, 3D range, infrared, color and polarized data, for a diverse range of applications such as industrial inspection, target tracking and wide field surveillance.

The LVS system supports both Windows 95/NT and Linux development. Linux supports the IUE development on the LVS and allows easy remote access/monitoring, and in most cases, superior performance. Windows provides familiar interface and drivers for a wider array of specialized devices. We are currently using IBM Thinkpads for the LVS. One LVS is an Thinkpad 760ed which has an integrated frame-grabber and provides self contained operation at 5-10Hz. We also have an LVS using 755cx using a PCMCIA frame-grabber, which is slower and not quite as accurate as the 760ed-based system. For problems demanding higher performance we are mating the 760 laptop with a docking station containing a Matrox Meteor PCI frame-grabber.[1] The LVS is outfitted with wireless PCMCIA networking for total mobility. Though it costs more than the desktop, the LVS offers the attraction of being mobile for use in battlefield and on factory floor.

The LVS interfaces either directly through its own parallel port, or indirectly over the network to ex-

[1] The very inexpensive CMOS camera's such as a quickcam are easily adapted to the system, but the quality of their imaging is usually not sufficient for industrial, surveillance or research needs.

ternal hardware such as a rotary table, the Active Eye/SPM-based pointing system and various other actuators. We are now evaluating the added benefit of the new MMX instruction set The initial evaluations show that the CPU cycles of common image processing operations can be reduced by a considerable amount. For example, using optimized MMX code (from Intel corporation) can increase 3x3 median filtering by a factor of 4. When data is restricted to 16bit integers, matrix transpose (up to 1024x1024 matrix) is about twice as fast, and matrix and vector multiplication (up to 2.5M elements) is 10 times faster. We expect this to be important for our applications as the key step in the SLAM software is exactly a matrix vector product.

We have built a portable environment for near real time vision and image processing using low cost components. The development of vision and image processing algorithms is split between IUE development and small stand alone "windows" programs encapsulating already developed ideas. We have looked into the possibility of achieving real time image processing using optimized software, and identified an application (SLAM) for evaluating various issues since SLAM itself is applicable to disparate sensor sources; it is also generic and powerful enough for solving a diverse range of vision problems real time with its simple and efficient matching technique.

Summary

This multi-disciplinary project has produced results in advanced sensor development, sensor planing, sensor fusion, vision module development, vision/robotic interfaces and vision software systems. We have produced significant results in these areas and in so doing advance the state of the art of visual sensor systems for manufacturing and for battlefield awareness. Our future goals are to continue to build on the science we have laid in the past two years, to evaluate the component technologies we have developed and to transition more of our work into manufacturing practice.

References

Note: References in bold-face were produced under our DARPA/ONR/MURI support.

[Abrams *et al.*, **96]** S. Abrams, P.K. Allen and K. Tarabanis. Computing camera viewpoints in a robot work-cell. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 1972–1979, Apr. 22-25 1996.

[Abrams, 97] S. Abrams. *Sensor Planning in an active robot work-cell.* PhD thesis, Dept. of Computer Science, Columbia University, Jan. 1997.

[Allen *et al.*, **96a]** P.K. Allen, A. Miller, P. Oh, and B. Leibowitz. Integration of vision and force sensors for grasping. In *Proc. Multi-Sensor Fusion and Integration*, pages 349–356, Dec. 9-12 1996.

[Allen *et al.*, **96b]** P.K. Allen, B. Yoshimi, A. Miller, P. Oh, and B. Leibowitz. Visual control for robotic hand-eye coordination. In *Workshop on Robotics and Robotic Vision, IEEE Int. Symp. on Signal Processing and Applications*, pages 20–37, Aug. 25-30 1996.

[Allen *et al.*, **97]** P.K. Allen, A. Miller, P. Oh, and B. Leibowitz. Using tactile and visual sensing with a robotic hand. In *IEEE Int. Conf. on Robotics and Automation*, April 22-25 1997.

[Baker and Nayar, 96] S. Baker and S.K. Nayar. Pattern rejection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 544–549, 1996.

[Baker *et al.*, **98a]** S. Baker, S.K. Nayar and H. Murase. Parametric feature detection. *Proc. of DARPA Image Understanding Workshop*, to appear, 1998.

[Baker *et al.*, **98b]** S. Baker, S.K. Nayar and H. Murase. Parametric feature detection. *Int. Journal of Computer Vision*, to appear 1998.

[Bascle *et al.*, 96] B. Bascle, A. Blake and A. Zisserman. Motion deblurring and super-resolution from an image sequence. *Computer Vision—ECCV*, pages 573–581, Apr 1996.

[Bhat and Nayar, 96] D. N. Bhat and S. K. Nayar. Ordinal measures for visual correspondence. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 351–357, 1996.

[Bhat *et al.*, **97]** D. N. Bhat, S. K. Nayar and A. Gupta. Motion estimation uisng ordinal measures. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 1997. To appear.

[Blum *et al.*, **97]** R. S. Blum, S. A. Kassam and H. V. Poor. Distributed detection with multiple sensors: part ii - advanced topics. *Proc. of the IEEE*, pages 64–79, Jan 1997.

[Blum, 95] R. S. Blum. Data fusion for manufacturing,. In *Proc. of the Information Resources in Manufacturing, 9th Conf. with Industry,*, Lehigh Univ., Bethlehem, PA,, May 1995.

[Califano and Mohan, 91] A. Califano and R. Mohan. Multidimensional indexing for recognizing visual shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 28–34, June 1991.

[Canny, 86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Chiang and Boult, 96a] M.C. Chiang and T.E. Boult. Efficient image warping and super-resolution. *Proc. of the Third IEEE Workshop on Applications of Computer Vision*, Dec 1996.

[Chiang and Boult, 96b] M.C. Chiang and T.E. Boult. The integrating resampler and efficient image warping. *Proc. of the ARPA Image Understanding Workshop*, pages 843–849, Feb 1996.

[Chiang and Boult, 97a] M.C. Chiang and T.E. Boult. Local blur estimation and super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition*, 1997. To appear.

[Chiang and Boult, 97b] M.C. Chiang and T.E. Boult. Imaging-consistent super-resolution. In *Proc. of the DARPA IUW*, 1997. These proceedings.

[Currie *et al.*, 95] N. C. Currie, F.J. Demma, D. D. Ferris Jr., R.W. McMillan, V.C. Vannicola, and M.C. Wicks. A survey of state-of-the art technology in remote concealed weapon detection. Technical report, Rome Laboratory, Rome NY, 1995.

[Currie *et al.*, 96] N. C. Currie, F.J. Demma, D. D. Ferris Jr., R.W. McMillan, V.C. Vannicola, and M.C. Wicks. Darpa/nij/rome lab. concealed weapon detection program: An overview. In *Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, pages 492–502, Orlando, FL, 1996. SPIE.

[Dana *et al.*, 97] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real world surfaces. In *Proc. IEEE Computer Vision and Pattern Recognition*, 1997. to appear.

[Deans and Blum, 96] M. C. Deans and R. S. Blum. Distributed signal detection system design using adaptive signal processing techniques. In *29th Conf. on Information Sciences and Systems*, pages 1065–1070, Princeton, Mar 1996.

[Deans, 96] M. Deans. An adaptive algorithm for the design of distributed detection systems. Master's thesis, Lehigh University, May 1996.

[Gideon and Hollister, 87] R. A. Gideon and R. A. Hollister. A rank correlation coefficient. *Journal of the American Statistical Association*, 82(398):656–666, 1987.

[Guttman, 84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD*, pages 47–57, June 1984.

[Huang and Tsai, 84] T. S. Huang and R. Y. Tsai. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, 1:317–339, 1984.

[Irani and Peleg, 91] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53(3):231–239, May 1991.

[Irani and Peleg, 93] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, Dec 1993.

[Kjeldsen and Kender, 96a] R Kjeldsen and J.R.Kender. Finding skin in color images. In *Proc. of the Second Int. Conf. on Automatic Face and Gesture Recognition*, pages 312–317, October 1996.

[Kjeldsen and Kender, 96b] R Kjeldsen and J.R.Kender. Toward the use of gesture in traditional user interfaces. In *Proc. of the Second Int. Conf. on Automatic Face and Gesture Recognition*, pages 151–156, October 1996.

[Koenderink *et al.*, 96] J.J. Koenderink, A.J. van Doorn and M. Stavridi. Bidirectional reflection distribution function expressed in terms of surface scattering modes. In *Proc. European Conf. on Computer Vision*, pages 28–39, 1996.

[Murase and Nayar, 95a] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *Int. Journal of Computer Vision*, 14(1):5–24, January 1995.

[Murase and Nayar, 95b] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *Int. J. of Comp. Vision*, 14(1):5–24, 1995.

[Nalwa and Binford, 86] V.S. Nalwa and T.O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:699–714, 1986.

[Nayar and Baker, 97] S. K. Nayar and S. Baker. Complete Class of Catadioptric Cameras. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Nayar and Oren, 95] S. K. Nayar and M. Oren. Visual appearance of matte surfaces. *SCIENCE*, 267:1153–1156, Feb 1995.

[Nayar and Peri, 97] S. K. Nayar and V. Peri. Omnidirectional Video Cameras for Vision Based Graphics. Technical report, Dept. of Computer Science, Columbia University, U.S.A., January 1997.

[Nayar *et al.*, 94] S. K. Nayar, H. Murase and S. A. Nene. Learning, positioning, and tracking visual appearance. *Proc. of IEEE Int'l. Conf. on Robotics and Automation*, May 1994.

[Nayar *et al.*, 95] S. K. Nayar, S. A. Nene and H. Murase. Real-time 100 object recognition system. Technical Report CUCS-021-95, Dept. of Comp. Sci., Columbia Univ., New York, NY, USA, September 1995.

[Nayar *et al.*, 96a] S. K. Nayar, S. A. Nene and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 12(5):750–758, October 1996.

[Nayar *et al.*, 96b] S.K. Nayar, S. Baker and H. Murase. Parametric feature detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 471–477, 1996.

[Nayar, 97] S. K. Nayar. Catadioptric Omnidirectional Video Camera. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Nene and Nayar, 85] S. A. Nene and S. K. Nayar. A simple algorithm for high dimensional nearest neighbor search. Technical Report CUCS-030-95, Columbia University, New York, NY, October 1985.

[Nene and Nayar, 96] S. A. Nene and S. K. Nayar. Closest point search in high dimensions. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, June 1996.

[Oren and Nayar, 95] M. Oren and S. K. Nayar. Generalization of the lambertian model and implications for machine vision. *Int. J. of Comp. Vision*, 14:227–251, 1995.

[Peri and Nayar, 97] V. Peri and S. K. Nayar. OMNIVIDEO: Generation of panoramic and perspective video from omnidirectional video,. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Reed and Allen, 97a] M. Reed and P. Allen. 3-d modeling from range imagery: An incremental method with a planning component. In *Int. Conf. on Advances in 3-D Digital Imaging and Modeling*, May 12-15 1997.

[Reed and Allen, 97b] M. Reed and P. Allen. A robotic system for 3-d model acquisition from multiple range images. In *IEEE Int. Conf. on Robotics and Automation*, April 22-25 1997.

[Reed *et al.*, 97a] M. Reed, P. Allen and I. Stamos. Automated model acquisition from range images with view planning. In *Computer Vision and Pattern Recognition*, June 16-20 1997.

[Reed *et al.*, 97b] M. Reed, P. Allen and I. Stamos. Solid model construction using meshes and volumes. In *Proc. DARPA Image Understanding Workshop*, May 12-14 1997.

[Shen, 97] Y. Shen. Neural network approach to the design of distributed detection systems. Master's thesis, Lehigh University, May 1997.

[Sundaram and Nayar, 97] H. Sundaram and S. K. Nayar. Are textureless scenes recoverable? In *Proc. IEEE Computer Vision and Pattern Recognition*, 1997. to appear.

[Tarabanis *et al.*, 94] K. Tarabanis, Roger Tsai and P.K. Allen. Analytical characterization of the feature detectability constraints of resolution, focus and field-of-view for vision sensor planning. *Computer Vision, Graphics, and Image Processing*, 59(3):340–358, May 1994.

[Tarabanis *et al.*, 95a] K. Tarabanis, R. Tsai and P. Allen. The MVP sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11(1):72–85, February 1995.

[Tarabanis *et al.*, 95b] K. Tarabanis, R. Tsai and P. Allen. Sensor planning in computer vision. *IEEE Trans. on Robotics and Automation*, 11(1):86–105, Feb 1995.

[Vikalo and Blum, 97a] H. Vikalo and R. S. Blum. Distributed detection in dependent non-Gaussian noise. In *Proc. IEEE Int. Sym. on Information Theory*, June 1997. to appear.

[Vikalo and Blum, 97b] H. Vikalo and R. S. Blum. Distributed detection of known signals in gaussian mixture noise which is dependent from sensor to sensor. In *Proc. Int. Conf. on Telecommunications*, 1997. to appear.

[Vikalo, 97] H. Vikalo. Distributed detection of known signals in impulsive noise,. Master's thesis, Lehigh University, Jan 1997.

[Watanabe and Nayar, 95] M. Watanabe and S. K. Nayar. Telecentric optics for constant magnification imaging. Technical Report CUCS-026-95, Dept. of Computer Science, Columbia University, New York, NY, USA, September 1995.

[Watanabe and Watanabe, 96] M. Watanabe and M. Watanabe. Minimal operator set for passive depth from defocus. *Proc. of IEEE CVPR*, June 1996.

[Watanabe and Watanabe, 98] M. Watanabe and M. Watanabe. rational filters for passive depth from defocus. *Int. Journal of Computer Vision*, 1998.

[Yoshimi and Allen, 97] B. Yoshimi and P.K. Allen. Integrating real-time vision and manipulation. In *Hawaii Int. Conf. on Systems and Science*, Jan. 8-10 1997.

[Zabih and Woodfill, 94] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. *Proc. of the European Conf. on Computer Vision*, pages 151–158, 1994.

[Zhang and Boult, 97] G. Zhang and T. Boult. Smart-pointers and data sharing in the iue. Technical report, Lehigh University, 1997.

[Zhong and Blum, 97] Z. Zhong and R. S. Blum. Multisensor image fusion using a region-based wavelet transform approach. In *Proc. of the DARPA IUW*, 1997. These proceedings.

[Zhong and Blum, 97] Z. Zhong and R. S. Blum. A region-based image fusion scheme for concealed weapon detection. In *Proc. 30th Conf. on Information Sciences and Systems*, Balt. MD, Mar 97.

[Zhou, 96] Yanhong Zhou. Research in applying image warping to polarization parameter fitting. Master's thesis, Dept. of EECS, Lehigh Univ., 1996. (Advisor: T. Boult).

Integrated Vision and Sensing for Human Sensory Augmentation

Takeo Kanade and Vladimir Brajovic
School of Computer Science, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
brajovic@cs.cmu.edu, tk@cs.cmu.edu
http://www.cs.cmu.edu/afs/cs/usr/brajovic/www/lab/vlsi.html

Abstract[1]

The Carnegie Mellon University MURI project sponsored by ONR performs multi–disciplinary research in integrating vision algorithms with sensing technology for low–power, low–latency, compact adaptive vision systems. These are crucial features necessary for augmenting the human sensory system and enabling sensory driven information delivery. The project spans four sub–areas ranging from low to high level of vision: (1) smart filters, based on the Acousto–Optic Tunable Filter (AOTF) technology; (2) computational sensor methodology, which integrates raw sensing and computation by means of VLSI technology; (3) neural-network based saliency identification techniques for identifying the most useful information for extraction and display; and (4) visual learning methods for automatic signal–to–symbol mapping.

1. Introduction

Automated vision and sensing research has made great strides in the last 30 years. Yet vision systems still lack attributes shared by most successful mass-market technologies — small size, low cost, low power and highly reliable performance. If computer vision processing had these characteristics, the potential applications would be nearly endless. Examples include: wearable smart vision systems for enhancing solder's situation awareness in the battlefield; head–up display vision enhancement systems for driving in bad weather and low visibility conditions; head–up display field telemedicine systems, and others. All these applications share common features — the applications are mobile and interact with the human sensory system. While today these scenarios are mostly futuristic speculations, some of the technologies they require have been partially demonstrated. Our research further develops these emerging technologies, and brings these visions closer to reality.

The CMU MURI project performs multi–disciplinary research spanning all levels of vision and sensing: dynamically tunable acousto–optic multispectral imaging [Brajovic and Kanade, 1997]; VLSI–based computational sensors [Brajovic and Kanade, 1997]; neural network saliency detection [Pomerleau, 1997]; automatic visual acquisition of object models [Hebert et al., 1997]; domain–independent evolution–based learning for signal–to–symbol mapping [Glickman and Sycara, 1997]; and learning coordination among multiple signal-to-symbol mapping agents [Teller and Veloso, 1997b]. We believe that the tight integration of vision algorithms and sensing technology will result in low–power, low–latency, compact, adaptive vision systems crucial for effective human sensory augmentation.

1.1. CMU Approach

The separation of sensing and processing, as a natural consequence of a conventional vision system comprising a camera and computer, results in several deficiencies. The two most critical features missing in this sens–and–process paradigm are *low latency processing* and *sensory adaptation.*

[1] This research has been sponsored in full or in part by Office of Naval Research (ONR) under Contract N00014-95-1-0591. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR or the U.S. Government.

Latency, or reaction time, is the time that a system takes to react to an event. The primary sources of latency in vision systems are the: *data transfer bottleneck* caused by the need to transfer an image from the camera to the processor, and the *computational load bottleneck* caused by the processor's inability to quickly handle a large amount of visual data. The detrimental effects of both bottlenecks scale–up with the image size. Often, the system "receives" the image data too late to cope with fast events or to provide sensory feedback to a human user. For example, during the frame time of a conventional camera, a person's gaze direction can shift by 18 degrees. To ensure that the viewer feels comfortable and natural in head–mounted display applications, for example, delays must be less than 10 to 20 msec.

Another aspect presently missing in machine vision is top–down sensory adaptation. Complex ad–hoc algorithms that try to extract relevant information from inadequate sensor data are inevitably unreliable. In fact, time and time again it has been observed that using the most appropriate sensing modality or setup allows recognition algorithms to be far simpler and more reliable. For example, the concept of active vision proposes to control the geometric parameters of the camera (e.g., pan, tilt, etc.) to improve the reliability of the perception [Aloimonos, 1992]. It has been shown that initially ill–posed problems can be solved after the top–down adaptation of the camera's pose has acquired new, more appropriate image data. However, adjusting geometric parameters is only one level at which adaptation can take place. Another example of adaptation is multi–spectral imaging, which can eliminate confusion by providing sensor images appropriate for the task. Acquisition of appropriate sensor bands adaptively, however, is often difficult since most multi–spectral imaging devices have fixed spectral sensitivity, while the appropriate wavelengths to process vary as conditions and the task change. Therefore, a system that can adjust its operation at all levels, even down to the point of sensing, would be far more adaptive than one that tries to cope with the variations at the "algorithmic" or "motoric" level alone.

The two major shortcomings of the sense–and–process approach which are outlined above, along with the fact that this approach naturally leads to bulkier and less cost-effective systems, suggest that an alternative is needed. We are establishing a new paradigm in which sensing and vision processing are tightly coupled for fast, time–critical, adaptive operation.

The following sections describe basic techniques and technologies that the CMU team has worked on; we believe these are necessary for the success of a low–latency adaptive vision system for human sensory augmentation.

2. Multi-Spectral Imaging Filters

Contributors: L.J.Denes, M. Gottlieb, B.Kaminsky, P. Metes, Z.K. Kun, M. Capizzi, J. Hibner, D. Purta, A.M. Guzman

This program task incorporates the spectral (color) dimension into the visual reasoning process. A programmable optical filter is utilized at the system's front end to reduce the computational load and its resulting bottlenecks in future automated vision systems. Filtering the incoming scene according to its spectral composition can remove a large amount of undesirable background clutter prior to higher level processing. Figure 1 is a schematic representation of the process. Enhanced performance is anticipated in a variety of applications, including human sensory augmentation systems for driver assistance. Because of its ability to extract and track objects, this vision system will more closely mimic the human observer.

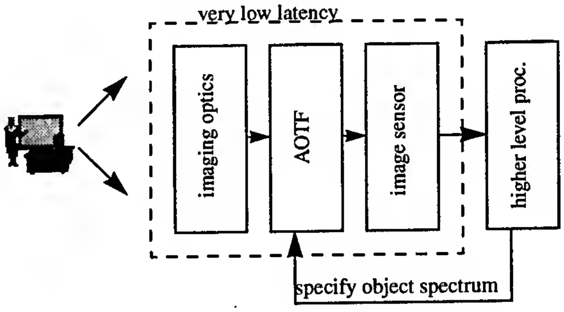


Figure 1: Object recognition using color discrimination.

We have assembled a multi-spectral imaging system operating in the visible to near IR range utilizing an existing acousto-optic tunable filter (AOTF). This configuration has been characterized, yielding design optimization information. Critical data include spatial and spectral resolution, out-of-band rejection, efficiency, field of view, and bandwidth. The design goal is efficient operation over nearly two octaves of wavelength, and superior image quality. Two major issues were successfully addressed. The first relates to the method of applying the multiple electrical RF control signals to the AOTF transducer to fully exploit the multispectral capability. Several approaches were analyzed, including multiple oscillators, spread spectrum techniques, and the use of an arbitrary waveform generator. Recent work has confirmed that the arbitrary waveform generator provides all of the flexibility required with no serious disadvantages. In addition, it is readily adaptable to computer control. The second issue addressed is how to best achieve object identification using color signature information. A fundamental issue arises because any background object with a broadband color distribution, e.g., white, will include the desired signature within its spectrum. Thus, these background objects may not be discriminated against the target object. To address this problem, we developed a processing technique using two video frames, in which the first frame grab contains a multispectral image whose spectral content lies outside the target color signature. This frame is inverted and then used as a spatial mask over the entire scene. The second frame grab includes only the target color signature and provides us with a gray scale. By using an appropriate threshold, the target image alone is displayed against a black background. Tests of laboratory scenes give encouragingly good results.

3. Computational Sensors for Low–Latency Adaptive Vision

Contributors: Vladimir Brajovic and Takeo Kanade

The computational sensor paradigm [Kanade and Bajcsy, 1993] has the potential to greatly reduce latency and provide top–down sensory adaptation to the vision system. By integrating sensing and processing on a VLSI chip, both transfer and computational bottlenecks can be alleviated: on–chip routing provides high throughput transfer; an on-chip processor could implement massively–parallel fine–grain computation providing high processing capacity which readily scales up with the image size. In addition, the tight coupling between processor and sensor allows for efficient top–down feedback that can control and adjust the sensor for further acquisition based on the preliminary results of the processing.

Our recent work has been concerned with efficient implementation of global operations over large groups of image data using a computational sensor paradigm [Brajovic and Kanade, 1994]. Global operations are important because: (1) in perception, each decision is a kind of global, or overall, conclusion necessary for the coherent interaction with the environment, and (2) unlike local operations (e.g., filtering) which produce large amounts of preprocessed image data, global operations produce *a few* quantities for the description of the environment which can be *quickly* transferred and/ or processed to produce an appropriate action for a machine. The main difficulty with implementing global operations comes from the necessity to bring together all or most of the data in the input data set. We have formulated two mechanisms for implementing global operations in computational sensors: (1) *sensory attention [Brajovic and Kanade, 1997]*, and (2) *intensity–to–time processing paradigm [Brajovic and Kanade, 1996]*.

The *sensory attention* is based on the premise that salient features within the retinal image represent important global features of the entire image. This premise is attractive for two reasons. First, the main argument that has been used to explain the need for selective visual attention in brains is that, as there exist some kind of processing and communication limitations in the visual system, the same exists in machines. Attention "funnels" only relevant information and protects the limited communication and processing resources from the information overload. Indeed, the importance of selecting the relevant information from an image is now widely acknowledged in machine vision; some forms of attention mechanisms (e.g. selecting a correctly sized window within the image) are often employed in practical applications. Second, it has been shown that the visual attention improves performance, and is needed for maintaining coherent behavior while interacting with the environment (i.e. attention–for–action) [Allport, 1989]. Location of such attention must be maintained in the environmental coordinates, thus maintaining coherence under ocular and head motion [Milanese, 1993]. Unlike eye movement (i.e., *overt* shifts), the attention shifts (i.e. *covert* shifts) do not require any motor action, but occur internally on a fixed retinal image. For this reason, attention shifts are faster and play an important role in low–latency vision systems.

We have implemented sensory attention by fabricating and testing a *tracking computational sensor.* This track sensor optically receives a saliency map and continuously selects and tracks the peaks in the map. The location and intensity of the selected saliency peaks is reported on few output pins with low latency. These quantities are also used internally in a top–down fashion to aid tracking of the attended location. The chip is a 28 x 28 array of 60μ x 60μ cells, and is fabricated on a 2.2mm x 2.2mm die. When tracing bright, well-defined features, the sensor tracks targets moving across the retina at about 6900 cells/second.

The *intensity–to–time processing paradigm* is based on the notion that stronger signals elicit responses before weaker ones, thus allowing a global processor to make decisions based on only a few inputs at a time. The key is that some preliminary decisions about the retinal image can be made as soon as the first responses are received. The intensity–to–time processing paradigm is used for the VLSI implementation of a *sorting computational sensor* — a sensor that sorts input stimuli by their intensity as they are being sensed. The chip detects an image focused thereon and computes an image of indices. During the computation, the chip computes a cumulative histogram — one global quantity of the detected image — and reports it with low-latency on one of the pins before the image is ever read out. The cumulative histogram is used internally in a top-down fashion to generate indices within each pixel. The image of indices has a uniform histogram which has several important properties: (1) the contrast is maximally enhanced, (2) the available dynamic range of readout circuitry is equally utilized, i.e., the values read out from the chip use available bits most efficiently, and (3) the image of indices never saturates, and preserves the same range (e.g., from 1 to N) under varying conditions in the environment.

The adaptation of the dynamic range of the sorting sensor is illustrated in Figure 2 showing sequence of 93 images provided by the sorting sensor. By observing the wall in the background, we can see the effects of adaptive dynamic range: even though the physical wall does not change the brightness, it appears dimmer in those frames in which bright levels are taken by pixels which are physically brighter (e.g., subject's face and arm). When the subject turns and fills the filed–of–view with dark objects (e.g., hair) the wall appears brighter since it is now taking higher indices. Also, note that the maximum contrast is maintained in all the images since all images of indices have uniform histogram.

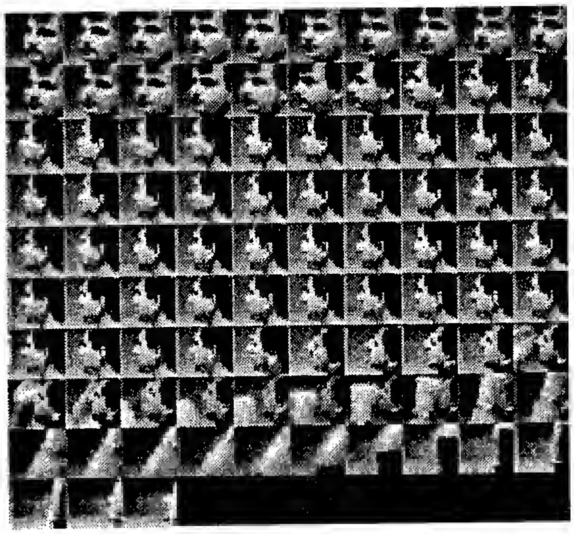


Figure 2: Sequence of images of indices computed by the sorting sensors.

We continue to work on an improved sorting computational sensor with smaller pixels and a larger array. We also continue to work on developing new sensors based on the intensity-to-time processing paradigm. We have designed, and recently received a prototype of, a self-contained eye tracking sensor. We plan to test the sensor and apply it in several scenarios. In the near term, we will begin interfacing some of our computational sensors with smart AOTF filters.

4. Visibility Estimation from a Moving Vehicle

Contributor: Dean Pomerleau

Reduced visibility caused by fog, rain, snow, darkness and glare is a frequent contributing factor to traffic accidents [Najm et al., 1995]. In fact, some of the most serious of all highway incidents, sometimes involving dozens or even hundreds of vehicles, occur when reduced visibility conditions result in a chain reaction of crashes. Technologies typically employed to estimate visibility include: transmissometers, which measure the transmittance of the atmosphere over a baseline distance; and nephelometers, which measure the scattering coefficient caused by suspended particles in an air sample [National Weather Service 1996]. Unfortunately, these systems suffer from several drawbacks as they are not always estimating visibility from the driver's point of view. The only way to automatically estimate the cumulative influence of these factors on the driver's ability to see potential obstacles ahead is to employ a sensing system which reasonably matches the driver's perceptual characteristics. We developed a system that accomplishes this match by using a CCD video camera pointing out the windshield of the vehicle, and processing the same features processed by the human driver to estimate visibility.

Manual visibility estimates are typically made by attempting to detect high contrast targets at various known distances. The farthest distance at which a target can be reliably detected is considered the visibility distance. Ideally, an automated visibility estimation system should work the same way. Unfortunately, it is very difficult to consistently find high contrast targets at various known ranges from a moving vehicle. Even the features that are supposed to be consistent on a roadway, the lane markings, vary greatly in their appearance, and are in fact frequently missing or obscured. The Rapidly Adapting Lateral Position Handler (RALPH) system [Pomerleau and Jochem, 1996] overcomes this difficulty when detecting the position and curvature of the road ahead in camera images by utilizing whatever features are visible on the roadway. These features may include lane markings, road/shoulder boundaries, tracks left by other vehicles, and even subtle pavement discolorations like the oil stripe down the lane center when necessary. Our visibility estimation system exploits RALPH's ability to find and track arbitrary road features. In short, the system estimates visibility by measuring the attenuation of contrast between consistent road features at various distances ahead of the vehicle.

The visibility estimation algorithm performs well under a wide variety of conditions. The rank ordering of six conditions tested corresponds reasonably well to one's intuitive notion of how difficult it is to see in these situations. Live vehicle tests in fog still need to be conducted (fog is rare in Pennsylvania, particularly during the winter when these experiments were conducted). However, the results from the simulated fog experiments and the live daytime tests in rainy conditions suggest that the algorithm should perform well, and report significantly reduced visibility under foggy conditions.

While all the work reported here has been done with a standard black and white CCD camera, we are investigating the potential for using alternative sensors for improved performance. For example, a high-dynamic range camera, such as a VLSI sorting computational sensor, would respond more like the human eye in extreme lighting conditions, and could therefore provide better visibility estimates. Another possibility would be to combine this visibility estimation technique with smart AOTFs for multispectral imaging. By testing the visibility at different wavelengths, it may be possible to select the best wavelength(s) for operation under the current conditions.

5. Multi-Agent Learning for Signal Classification in Vision

Contributors: Astro Teller and Manuela Veloso

A wide variety of machine learning mechanisms create multiple models that must be reconciled, chosen among, or in some cases, *orchestrated.* In its most general form, this orchestration problem can be seen as part of the multi-agent learning problem.

There are many cases in which a task to be approached with machine learning techniques can be, or must be, solved in more that one "piece." Learning a team of robotic soccer players is a good example of a task that could conceivably be done as a single agent, but lends itself very naturally toward learning sub-solutions and *then* (or in addition) learning to ensure the mutual suitability of these sub-solutions. This insurance of mutual suitability is the *orchestration problem.*

Evolutionary computation is a natural machine learning environment in which to find many, behaviorally distinct models. We focus on PADO, a evolutionary computation framework designed specifically for signal classification (e.g., [Teller and Veloso, 1997b]). As a process of divide and conquer, PADO evolves multiple pools of sub-solutions and then orchestrates one or more learned models from each pool.

The question we investigate is: "What opportunities are there for learning in the orchestration process and how much improvement can this learning provide?" While answering this question, our research demonstrated several things [Teller and Veloso, 1997b]. First, specific experiments on distinct signals demonstrated the feasibility of PADO's divide and conquer strategy; the failure of the evolved orchestration procedure suggested PADO's preferability to unconstrained learning. Second, the experiments provided a specific justification for maintaining a population; orchestration puts the options a population provides to good use. And finally, this work introduced specific techniques for orchestration learning and, through their successful application, demonstrated that orchestration is an important issue and that learned orchestration can provide dramatic generalization improvements.

6. Adaptive Acquisition of Search Control Knowledge in the Evolution of Face Recognition Neural Networks

Contributors: Matthew Glickman and Katia Sycara

Search algorithms for signal–to–symbol matching patterned after biological evolution are attractive for use in domains such as vision that have complex search spaces for a number of reasons. These include: (1) Their application does not explicitly require deep insight into the domain; (2) They are relatively straightforward to paralyze; and (3) their natural analog has resulted in entities of extraordinary complexity and robustness. However, the search performance in any particular domain is highly dependent on the interaction between the chosen representation of the space and the specific search operators employed. For evolutionary algorithms in particular, this interaction is a poorly understood process, leaving practitioners with few guidelines as to how to make the right choices to yield good performance.

One popular approach to improving the performance of search in a particular domain is to seek to incorporate pre-existing knowledge of the domain into the operators and representation. However, this approach is problematic for evolutionary search because of the aforementioned opacity of the interaction between the operators and the representation. This difficulty, popularly known as ``the representation problem," is only compounded in more complex domains, presenting a formidable obstacle to the application of artificial evolution in precisely those domains in which they may be of the greatest utility.

Therefore, rather than seeking to find how pre-existing domain knowledge can be best exploited by evolution, our research is directed toward the automatic acquisition of such knowledge in operational form. The experiments reported herein demonstrate that information about a particular domain generated over the course of evolutionary search can be extracted, analyzed, and then employed to improve search in future runs.

The space explored is the weight space of fixed-topology, feed-forward artificial neural networks (ANNs) for face recognition. Over the course of adaptation, weight vectors, along with their self-adapted, variable mutation rate, were collected. These data were then used to train another ANN to predict the appropriate mutation rate for a given weight vector for the face-recognition domain in general. Finally the mutation rate-prediction networks were used to drive evolution on another face recognition task, resulting in networks with improved generalization performance.

Our preliminary results indicate that this approach is reliably feasible. Due to the fact that (1) the specific weight-vector/mutation-rate pairs chosen for training were selected via a simple, Darwinian selection process, and (2) that the target mutation rates contained in these data had also been adapted via this same selection process, the results reported here indicate that simple Darwinian selection is sufficient to generate a training signal from which domain/search-control knowledge may be extracted. This result indicates a promising direction for the successful application of artificial evolution in complex domains such as image understanding.

7. Visual Learning for Landmark Recognition

Contributors: Martial Hebert, Katsushi Ikeuchi, Yukata Takeuchi, Patrick Gros

Recognizing landmarks is a critical task for interaction of a machine with the environment. Landmarks are used for building maps of unknown environments. In this context, the traditional recognition techniques based on strong geometric models cannot be used. Rather, models of landmarks must be built from observations obtained using image-based techniques. This section describes building image-based landmark descriptions from sequences of images, and then recognizing the landmarks. This approach also addresses the more general problem of identifying groups of images with common attributes in sequences of images. We show that, with the appropriate domain constraints and image descriptions, this can be done using efficient algorithms.

Recognizing landmarks in sequences of images is a challenging problem for a number of reasons. The appearance of any given landmark varies substantially from one observation to the next. In addition, to variation due to different aspects, illumination change, external clutter, and changing geometry of the imaging devices are other factors affecting the variability of the observed landmarks. Finally, it is typically difficult to use accurate 3D information in landmark recognition applications. For those reasons, it is not possible to use many of the object recognition techniques based on strong geometric models.

The alternative is to use image-based techniques in which landmarks are represented by collecting images which are supposed to capture the "typical" appearance of the object. The information most relevant to recognition is extracted from the collection of raw images and used as the model for recognition. This process is often referred to as "visual learning."

Progress has been made recently in developing such approaches. For example, in object modeling [Gross et al.], 2D or 3D model of objects are built for recognition applications. An object model is built by extracting features from a collection of observations. The most significant features are extracted for the entire set and are used in the model representation. Extensions to generic object recognition were presented recently [Carlsson, 1996].

Other recent approaches use the images directly to extract a small set of characteristic object images which are compared with observed views at recognition time. For example, the eigen-images techniques are based on this idea.

Those approaches are typically used for building models of a single object observed in isolation. In the case of landmark recognition for navigation, there is no practical way to isolate the object in order to build models. Worse, it is often not known in advance which of the objects observed in the environment would constitute good landmarks. Visual learning must therefore be able to identify groups of images corresponding to "interesting" landmarks and to construct models amenable to recognition out of raw sequences of images.

A similar problem, although in a completely different context, is encountered in image indexing, where the main problem is to store and organize images to facilitate their retrieval [Lamiroy and Gros, 1996] [Schmid and Mohr, 1996]. The emphasis in this case is on the kind of features used and the types of requests that can be made by the user. For image retrieval, actual systems (QBIC, JACOB, Virage...) are closer to smart browsing than to image recognition. Using criteria such as color, shape, regions, etc., the systems search for images most similar to a given image. The user can then interact with the system to define which of these images seems the most interesting, and a new set of closer images is displayed.

Our system tries to combine those two categories of systems. In a training stage, the system is given a set of images in sequence. The aim of the training is to organize these images into groups based on similarity of feature distributions between images. The size of the groups obtained may be defined by the user, or by the system itself. In the latter case, the system tries to find the most relevant groups, taking the global distribution of the images into account. In a second step, the system is given new images, which it tries to classify as either one of the learned groups, or belonging to the category of unrecognized images. Figure 4 shows indentifying landmarks from a moving vehicle.

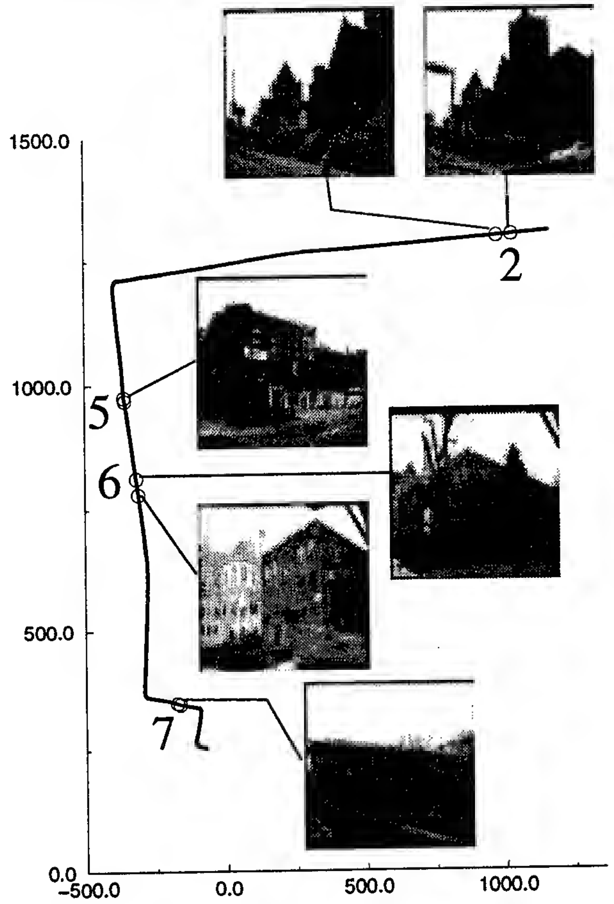


Figure 3: Overhead view of the path followed while collecting the images.(distances are indicated in meters.) Four landmarks are correctly identified, corresponding to groups 2, 5, 6, and 7 of the training sequence. Example images from the test sequence are shown for

The basic representation is based on distributions of different feature characteristics. All these different kinds of histograms are computed for the whole image and for a set of sub-images. Tests similar to Chi-square tests are used to compare these histograms and define a distance between images. This distance is then used to cluster the images in what are called groups. An agglomerative grouping algorithm is used at this stage. At each step of the algorithm, the clusters made are evaluated by an entropy-like function, whose maximum gives the optimal solution in a sense specified later. Each group is then characterized by a set of feature histograms. When new images are given to the system, it evaluates a distance between these images and the groups. The system determines to which group this image is the closest, and a set of thresholds is used to decide if the image belongs to this group.

The main goal of the work presented here was to explore the use of tools and methods in the field of image retrieval when applied to the problem of landmark recognition. It is clear that the global architecture of the system is close to that of object recognition systems [Gross et al.]: a training stage in which 3D shape, 2D aspects, or groups, are characterized is followed by a recognition stage in which this information is used to recognize the models, objects or groups in new images. The difference comes from the wide diversity of the images and from the groups which are not reduced to a single aspect of an object. The two challenging tasks which we concentrate on describing in the remainder of the paper are to define these groups more precisely as sets of images, and to automatically learn a characterization for each group: what remains invariant, what varies, and in which proportions.

8. Conclusion

CMU MURI performs cross –disciplinary research which will result in high performance vision systems adequate for "natural" human sensory augmentation and sensor driven information delivery. We are demonstrating progress in all levels of vision: from image formation and computational sensing to high level adaptive context–independent learning strategies. We believe that the tight integration of these techniques will provide opportunity for more efficient bottom–up and top–down control in vision processes which will result in low–power, low–latency, compact, reliable and

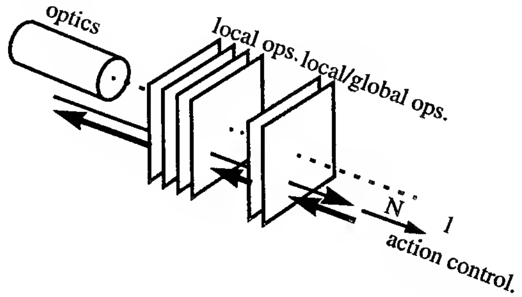


Figure 4: A vision system with tight integration of image formation, sensing and processing for adaptive low–latency applications.

adaptive vision systems (see Figure 4) crucial for effective human sensory augmentation.

References

[Allport, 1989] Allport, A. "Visual Attention", *Foundation of Cognitive Science,* M. Posner (ed.), MIT Press, 1989, pp. 631–682.

[Aloimonos, 1992] Aloimonos, J. (ed.), *Special Issue on Purposive, Qualitative, Active Vision,* CVGIP: Image Understanding, Vol. 56, No. 1, 1992.

[Brajovic and Kanade, 1994] Brajovic, V. and T. Kanade, "Computational Sensors for Global Operations", *IUS Proceedings*, pp. 621-630, 1994.

[Brajovic and Kanade, 1996] V. Brajovic and T. Kanade: "A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996, pp. 1638–1643.

[Brajovic and Kanade, 1997] Brajovic, V and T. Kanade: "Computational Sensors for Low–Latency Adaptive Vision," in IUW Proc., 1997.

[Carlsson, 1996] Carlsson, S. "Combinatorial Geometry for Shape Representation and Indexing," *Proc. International Workshop on Object Representation for Computer Vision.* Cambridge, England, April 1996.

[Brajovic and Kanade, 1997] Denes, L.J., M. Gottlieb, B.Kaminsky, P. Metes, Z.K. Kun, M. Capizzi, J. Hibner, D. Purta, and A.M. Guzman: "Multi-Spectral Imaging Filters," in IUW Proc., 1997.

[Glickman and Sycara, 1997] Glickman, M. and K. Sycara: "Adaptive Acquisition of Search Control Knowledge in the Evolution of Face Recognition Neural Networks," in IUW Proc., 1997.

[Gross et al.] P. Gros, O. Bournez and E. Boyer. *Using Local Planar Geometric Invariants to Match and Model Images of Line Segments.* To appear in Int. J. of Computer Vision and Image Understanding.

[Hebert et al., 1997] Hebert, M., K. Ikeuchi, Y. Takeuchi, P. Gros, "Visual Learning for Landmark Recognition," in IUW Proc., 1997.

[Kanade and Bajcsy, 1993] Kanade, T. and R. Bajcsy, "Computational Sensors: A Report from DARPA workshop", *IUS Proceedings*, 1993.

[Lamiroy and Gros, 1996] B.Lamiroy and P.Gros. Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing. *Proc. of the 4th European Conf. on Computer Vision*, Cambridge, England, pages 59--70, Vol. 1, April 1996.

[Milanese, 1993] R. Milanese, *"Detecting Salient regions in an Image: from Biological evidence to computer implementation"*, Ph.D. Thesis, Dept. of Computer Science, U. of Genova, Switzerland, December 1993.

[Najm et al., 1995] Najm, W., Mironer, M. and Fraser, L. (1995) "Analysis of Target Crashes and ITS Countermeasure Actions". *Proc. of 1995 ITS America Annual Meeting,* pp. 931-940.

[National Weather Service 1996] "*Surface Weather Observations and Reports*" (1996) Federal Meteorological Handbook, 5th Edition, National Weather Service Publication FMH-1.

[Pomerleau and Jochem, 1996] Pomerleau, D. and Jochem, T. (1996) Rapidly Adapting Machine Vision for Automated Vehicle Steering. *IEEE Expert, Vol. 11, No. 2.* pp. 19-27.

[Pomerleau, 1997] Pomerleau, D. "Visibility Estimation from a Moving Vehicle Using the RALPH Vision System", in IUW Proc., 1997.

[Schmid and Mohr, 1996] C. Schmid and R. Mohr. Combining Greyvalue Invariants with Local Constraints for Object Recognition. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA. pages 872--877, June 1996

[Teller and Veloso, 1997b] Teller, A. and M. Veloso, "Learning Better Classification Teams," in IUW Proc., 1997.

Principal Investigator Report: Automated Vision and Sensing Systems at Boston University *

Stephen Grossberg, Gail Carpenter, Eric Schwartz, Ennio Mingolla,
Daniel Bullock, and Paolo Gaudiano,
Department of Cognitive and Neural Systems, Boston University;
Andreas Andreou and Gert Cauwenberghs, Department of Computer Science
and Electrical Engineering, Johns Hopkins University; and
Allyn Hubbard, Department of Electrical and Computer Engineering, Boston University
email: `steve@cns.bu.edu` URL: `http://cns-web.bu.edu/muri`

Abstract

Our Center will continue to develop general-purpose autonomous systems for vision, object recognition, and control applications. The systems are realized in software, off-the-shelf hardware, and customized chips. These systems are designed to operate within noisy environments for which rules are not known and which can change unexpectedly through time. They typically begin with models of a key brain competence and end with fielded applications that have been thoroughly benchmarked. The design of adaptive algorithms will be emphasized, as will transfer of well-characterized algorithms to a larger class of applications and to real-time platforms. New projects will continue to include psychophysical studies of how humans search complex scenes; models of coherent processing of noisy and incomplete image data from natural and artificial sensors; development of self-organizing classifiers capable of fast, stable, distributed, incremental learning and hypothesis testing in response to nonstationary, incomplete, and probabilistic data; algorithm and hardware development for head-mounted space-variant active vision systems; development of self-calibrating autonomous robots; and fabrication of chips for vision and classification applications.

1 Introduction

This report summarizes new research projects to be conducted under the Multidisciplinary University Research Initiative (MURI) program by the Boston University Department of Cognitive and Neural Systems, the Boston University College of Engineering, and the Johns Hopkins University Department of Electrical and Computer Engineering. Our companion Technical Report [Grossberg *et al.*, 1997] reviews our research approach and some of our current efforts to develop general-purpose autonomous neural systems for vision, object recognition and control applications. The present PI report briefly lists the objectives, research questions, and evaluation procedures that will be used in our continuing work on these and related projects. Background information should be sought in the Technical Report.

2 Boundary and Surface Processing of Natural and Synthetic Images (Investigators: A. Baloch, S. Grossberg, R. Raizada, J. Williamson)

Objectives: To continue development of boundary segmentation and surface representation algorithms to enhance image data from synthetic aperture radar (SAR), multispectral infrared (IR), laser detection and ranging (LADAR), and related sensors for use by expert photointerpreters, by non-expert users in battlefield conditions, and as a preprocessor for

* Supported in part by the Defense Advanced Research Projects Agency and the Office of Naval Research (ONR N00014-95-1-0409).

such image data before it is automatically classified by adaptive pattern recognition algorithms. These results may thus be used for a wide variety of image exploitation, visual surveillance, and geospatial modeling applications.

Research Questions: Two types of research issues will be emphasized. The first concerns how these boundary segmentation and surface representation circuits can self-organize their own optimal operating parameters. Such a development would be important from at least two perspectives: It would greatly reduce the time needed to design such a circuit by allowing the circuit itself to discover its best operating parameters. It would also enable the circuit to autonomously recruit processing resources in response to changing environmental statistics to better discriminate data features that may have not been anticipated in its original design. The second research issue concerns how to realize this latest generation of algorithms in commercial off-the shelf digital signal processing boards that can run in real-time, and to work with our VLSI (very large scale integrated circuits) teams at Johns Hopkins and Boston University to begin implementing them in compact, lightweight, low-power, real-time chips.

As noted in Grossberg *et al.* [1997], the circuits in question have been derived from an analysis of how the visual cortex carries out similar tasks. It is well-known from neurobiological experiments that these cortical circuits may be tuned by visual experience. This is true both for the adaptive filters that regulate the bottom-up flow of signal processing, and for the horizontal connections that subserve the boundary segmentation process. We will investigate how both types of circuits may learn their own best operating parameters.

To implement these algorithms in off-the-shelf digital signal processing boards, we will investigate, among others, the Texas Instrument TMS320C80, which has been designed for imaging and graphics applications. The C80 combines 4 digital signal processors, a RISC processor, and an I/O controller on a single chip optimized for bandwidth-intensive applications requiring massive parallel processing. The C80 will be adapted for real-time implementation of neural systems for processing both static scenes and attentive motion grouping applications, as noted in Section 6.

Evaluation: The self-organization project will first be evaluated by using the new adaptive cortical circuit to explain and simulate key neurophysiological data about the timed formation of identified cortical connections. When the biological version of the model is finished, it will be used, as in the case of the non-adaptive model, to enhance SAR data. First, the adaptive model will be trained on SAR data to study how the statistical properties of these data determine the best filter and grouping parameters through learning. This self-organized circuit will then be benchmarked against the hand-crafted circuits that have previously been used. Finally, in later years, the circuit will be trained and tested on data from other sensors to understand how the self-organized parameters for each sensor type may differ, and to develop a strategy for implementing a general-purpose boundary and surface processing system with these results as a guide.

The circuit board implementation project will acquire a COTS C80/PCI powered board with associated hardware and software, and immediately study how to achieve maximum scalability of such boards for future projects. Software libraries for basic circuit modules of the above architectures will then be developed. These modules will be integrated into architectures for processing textured scenes. The results derived with the real-time hardware will be compared with those derived on present versions of the models.

In addition to these research projects, we will also continue to transfer this technology as it develops to users like the Machine Intelligence Group at MIT Lincoln Laboratory, for testing on their classified DoD data.

3 ARTEX Classifiers of 3-D Objects and Textured Scenes (Investigators: J. Brown, S. Grossberg, J. Williamson)

Objectives: There are three main objectives of this project: (1) To develop a version of Gaussian ARTMAP that uses more local computations, and thus one that should be more computationally efficient and embeddable within a larger architecture for high-level IU. (2) To implement the ARTEX architecture for classification of textured regions in a scene, notably from SAR and other military sensors, in real-time commercial off-the-shelf hardware and in custom VLSI. These results should provide real-time recognition of both natural and man-made objects that are detected by a wide variety of platforms. (3) To embed the ARTEX architecture into a larger architecture for recognition also of 3-D objects and more complex scenic configurations.

Research Questions: Our approach to the first two problems will follow the same format as in Section 2.

Evaluation: The projects in Section 2 will provide an improved front-end for the family of Gaussian ARTMAP classifiers that will be further developed in this project. In addition, the new Gaussian ARTMAP algorithms will be benchmarked against the Expectation Maximization (EM), rule-based, multilayer perceptron, and K-NN algorithms that were useful in our study of the previous version. The larger ARTEX architecture will be incorporated into a previously benchmarked VIEWNET architecture for incrementally learning to recognize 3-D objects from sequences of their 2-D views [Bradski and Grossberg, 1995]. This version of the VIEWNET architecture will be fitted with a new What-and-Where invariance filter [Carpenter, Grossberg, and Lesher, 1997] which will enable the system as a whole to incrementally learn predictive relationships among the objects of a scene and their spatial locations.

4 Model of 3-D Vision and Figure-Ground Separation (Investigators: S. Grossberg, F. Kelly, R. Paine)

Objectives: To further develop the FACADE model of 3-D vision and to apply it to more realistic scenes in which overlapping occluding and partially occluded objects occur. As noted in Grossberg *et al.* [1997], such preprocessing of an image before it is input to a pattern classifier can lead to higher classification accuracy of overlapping objects in the types of cluttered scenes that occur in the battlefield.

Research Questions: Research will focus upon how the multiple scales of the FACADE model work together to generate 3-D boundary and surface representations that correctly parse increasingly complex 2-D images and 3-D scenes into depthful surface representations. The completion of both boundary and surface representations of partially occluded objects for purposes of pattern recognition will also be emphasized.

Evaluation: The FACADE model will be further tested and developed by simulating key psychophysical examples of how human observers do 3-D figure-ground separation. A large number of displays will be simulated to derive multiple constraints on the circuit design. These simulations will include pop-out of occluding figures and amodal completion of occluded figures in response to line drawings, to surface renderings wherein the contrast relationships between abutting regions are given arbitrary relative values, to displays in which either transparent or opaque occlusion percepts can obtain, to displays in which relative brightness can in itself cause pop-out, and to ambiguous stratification displays in which bistable reversals of occluding and occluded surfaces occurs. After these psychophysical displays are simulated, we will begin to evaluate the model's pop-out and amodal completion properties on images derived from military sensors.

5 ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction

5.1 Geospatial Mapping (Investigators: G.A. Carpenter, J. Franklin, S. Gopal, S. Macomber, S. Martens, C. Woodcock)

Objectives: A remote sensing testbed allows performance comparisons between candidate neural network systems and state-of-the-art image processing and recognition techniques, in a collaborative project with researchers at the Boston University Center for Remote Sensing.

Research Questions: Phase 1 of this project developed fuzzy ARTMAP networks that were specialized to identify vegetation classes based on input that includes 6 spectral bands (visible and IR) and terrain variables, such as aspect, slope, and elevation. Researchers working on the new NASA Earth Observing System (EOS), are now using this work to help design systems for image analysis, data compression, feature extraction, and temporal prediction, to be placed in satellites scheduled for launch in 1998 and beyond. Phase 2 of the remote sensing project began with the planning of data collection for a study in the Plumas National Forest, in northern California, resulting in a rich ground truth data set that is now the basis for ongoing comparative studies. This project is currently developing automatic methods for mixture prediction for mapping tasks in which sites typically feature a composite of output classes, such as a vegetation mixture, rather than a single class. Researchers at MIT Lincoln Laboratory are already considering this mixture prediction method for remote sensing problems that range from finding tanks in infrared imagery to ecosystem mapping.

Evaluation: Phase 3 will use a larger database to develop, test, and field the results of the neural network prediction algorithms. New network hierarchies that take advantage of the database size and structure are also under development. The goal of this work is to produce systems that will rapidly provide accurate geospatial maps from high-dimensional satellite and terrain data. These methods could be applied for military intelligence as well as for the large-scale environmental mapping problems of the unclassified system development testbed.

5.2 Computer-Assisted Medical Diagnosis (Investigators: G.A. Carpenter, R. D'Agostino, J. Griffiths, N. Terrin)

Objectives: Medical database analysis has provided a fruitful medium in which to develop and test new learning algorithms, resulting in computational advances such as those of the ARTMAP-IC (instance counting) network.

Research Questions: A new opportunity for collaboration with medical statisticians and researchers at the New England Medical Center has recently arisen. This project, which will test and then field learning algorithms in medical settings, will give our researchers access to large-scale databases as well as cutting-edge biomedical expertise. Medical databases present many of the same challenges as might be found in other information management settings, including the battlefield, where speed, efficiency, ease of use, and accuracy are at a premium. In addition, a direct goal of improved computer-assisted medicine is to help deliver quality emergency care in situations that may be less than ideal, including those that military medical personnel might encounter.

Evaluation: Systems will be evaluated in terms of effectiveness in dealing with noisy, unreliable, and inconsistent data; nonstationary and region-specific variations; cases that are rare but significant; large-scale problems; and on-line learning situations. These are some of the issues that ARTMAP database analyses have been addressing in recent years, and new projects will continue to bring research prototypes closer to the implementation stage.

5.3 Sonar Target Recognition (Investigators: L. Burton, G.A. Carpenter, M.A. Rubin, W.W. Streilein)

Objectives: A new application project for sonar target recognition is now being developed. This work would be done in collaboration with researchers at the Orincon Corporation who are currently funded by a Phase II SBIR contract from the Office of Naval Research.

Research Questions: The project seeks to improve automated sonar recognition capabilities. Researchers at Orincon have, to date, examined the recognition capabilities of multilayer perceptrons and ellipsoidal basis functions, the latter giving slightly better results. Variations on these methods perform types of sensor fusion, presenting networks with both sets of processed data (SCAT and matched filter) or presenting a series of inputs taken from a series of aspects. The sonar recognition problem is similar to problems on which fuzzy ARTMAP and ART-EMAP (evidence MAP) have already proved successful. The data set would also be a useful testbed for enhanced ARTMAP recognition systems such as those being developed for radar recognition problems.

Evaluation: If pilot studies prove successful, large-scale implementation will follow. This type of collaborative interaction should accelerate transfer of new basic research results into applications technology by reducing duplicated effort, sharing software, and providing ongoing adaptation of basic neural network architectures to the particular demands of problems that are of direct interest to the military.

5.4 Radar Target Detection (Investigators: G.A. Carpenter, M.A. Rubin, W.W. Streilein)

Objectives: Comparative tests of buffering and other data compression methods on extended sets of simulated radar range profiles will continue in the coming year.

Evaluation: Further development of ARTMAP-FD (familiarity discrimination) will focus on automated methods for selecting the familiarity threshold. The aim is to produce a network that can perform on-line familiarity discrimination. Comparison with experiments on human familiarity discrimination may also suggest computational improvements to the neural network.

5.5 Fast Distributed Learning (Investigators: G.A. Carpenter, B. Noeske, W.W. Streilein)

Objectives: Basic research to develop the next generation of neural network pattern recognition devices will continue in parallel with technology transfer of more established algorithms.

Research Questions: One such project will investigate the new class of distributed ART models. Studies to investigate computational properties of distributed ART (dART) and distributed ARTMAP (dARTMAP) systems are leading toward systems that expand application capabilities of the ART family of networks. This project will develop a set of benchmark simulation examples to probe learning by distributed ARTMAP models. These examples will serve both to help guide design choices for networks that have already been specified and to suggest new networks with additional computational features.

Evaluation: Initial studies will examine low-dimensional problems in order to focus on details of system analysis and design. Later studies will include large-scale problems such as geospatial mapping and other applications.

6 Coherent Processing of Moving Targets (Investigators: A. Baloch, S. Grossberg, J. Richardson)

Objectives: To develop the Motion Boundary Contour System (mBCS) model for testing on complex imagery, and to begin to implement it on real-time hardware. As noted in our Technical Report, the mBCS model realizes a process of motion capture whereby objects that might not otherwise be detected in cluttered and scintillating scenes can pop-out coherently with a well-defined motion direction and speed. The model can also be attentively primed to selectively track objects which are moving in a prescribed direction. These properties are valuable for detecting and tracking objects under difficult sensing conditions.

Research Questions: A central research focus is to further develop our model of how form-and-motion information are fused together to detect and track moving objects. As noted in our Technical Report, when objects are detected by a small number of pixels, either due to their small size on the sensor or to interference by many noise pixels, an emergent boundary segmentation may be needed to detect them with high precision, before this segmentation is injected into the motion system at the proper processing stage. We will further develop the form-motion fusion model to do this in response to increasingly complex image sequences. Of particular importance is the question of how the motions of many densely moving objects are not confused with one another for purposes of target tracking.

Evaluation: The form-motion fusion model will be developed by simulating its responses to increasingly complex moving visual shapes, starting with filled-in forms, then outline forms, then forms defined by textures moving with respect to textured backgrounds with increasing levels of multiplicative noise. After these studies are completed, we will further develop and test the model on LADAR and multispectral IR imagery. Throughout this development process, we will also study how the types of processes that are used for form-motion fusion may be implemented on the C80 board.

7 Psychophysical Experiments for Real World Tasks

7.1 Visual Search Experiments in Cluttered Environments (Investigators: J. Beck, R. Cunningham, E. Mingolla)

Objectives: To help determine factors affecting human performance in visual searches for targets in clutter in grayscale imagery. Increasingly realistic scenery will be sought as mechanisms of visual search are better elucidated.

Research Question: One important factor is the lighting of natural scenes. Is the claimed human biases for interpretation of visual scenes as being illuminated from above true for a visual search task employing imagery more complex than often employed in the psychophysical laboratory?

Evaluation: Human performance for finding a target in clutter. on simple, computer-generated stimuli will be compared to that on more naturalistic imagery.

7.2 Experiments on the Perceived Segregation of Element-Arrangement Textures (Investigators: J. Beck, E. Mingolla, S. Oddo

Objectives: To help determine factors affecting human performance in breaking camouflage through perceptual segmentation of image regions from chromatic textural information.

Research Question: How do factors such as hue similarity, or the responses of retinal cone receptors, help to predict how readily a person can see textural differences in colored imagery?

Evaluation: Human performance on computer-generated stimuli has been measured in order to determine the fundamental mechanisms of human chromatic texture segregation.

8 Adaptive Control of Eye Movements (Investigators: G. Arakawa, D. Bullock, G. Gancarz, S. Grossberg)

Objectives: To integrate the multimodal fusion model for learning to compute ballistic movement decisions in response to competing visual, auditory, and planned movements into a larger system architecture for movement control. This system will be capable of controlling both ballistic and continuous tracking movements. The self-calibrating capability of this tracking system will be of value in many applications wherein system parameters may change due to use in the field. Psychophysical experiments will also be carried out to test concepts concerning how human observers continuously track moving targets under intermittently occluding conditions. These conditions will help to evaluate human performance under these conditions, as well as to design better automatic tracking systems for dealing with them.

Research Questions: Three central questions will be addressed: (1) How to combine ballistic and continuous tracking capabilities into a single self-calibrating system? In particular, how is the decision made as to whether continuous or ballistic tracking is released in a given context. (2) How to transform the decisions made within these systems into the controller which moves the self-calibrating motor plant? (3) How to set up the learning capabilities of the system so that all the different competing sensory sources can generate accurate movements, even though they use multiple converging and diverging pathways to move the tracking system.

Evaluation: We will continue to develop a model of how the brain accomplishes these feats. In particular, the ballistic and continuous movements to be modeled first are the saccadic and smooth pursuit movements of the eyes. A model of the interactions of the superior colliculus, frontal eye fields, and posterior parietal cortex for multimodal fusion will be joined to a model of how these regions interact with the cerebellum, reticular formation, and eye muscle plant to learn the correct gains whereby to generate accurate eye movements in response to all combinations of movement commands. The model will also be designed to explain how the human smooth pursuit system can adaptively maintain near-unity tracking gains by learning both to reconstruct target velocity and to compensate for sensory-motor processing lags. Experiments will be carried out to measure a precise estimate of the gain of smooth pursuit during an interval when a moving target is temporarily occluded. This experiment and others will be carried out on our recently purchased eye-head tracking system from ISCAN Inc., which will allow a full range of experiments to test models of eye-head cooperation in ballistic and continuous tracking.

9 Head Mounted Space-Variant Active Vision System: Algorithms and Hardware (Investigators: G. Bonmassar, B. Fischl, D. Fraser, E. Schwartz)

Objectives: To build miniature, low-cost, low-power, light weight, "wearable" space-variant active vision systems, and to provide algorithmic and hardware solutions towards this goal, implementing high performance visual navigation, surveillance, target acquisition, and pattern recognition. Applications include computer aided vision for low light, night vision, infra-red, visual prosthesis, and navigation.

Research Questions:

- Construction of wearable space-variant active vision system.
- Development of high-performance, real time image processing algorithms for early vision, navigation (e.g., by template matching to a geographic image database) and pattern recognition,(e.g., face recognition from a database of faces) for use in space-variant vision applications.
- Demonstration of man-machine interface demonstration enabling blind subjects to navigate and perform visual pattern recognition via the use of a wearable space-variant active vision system.
- Build and demonstrate hardware and electronics fabrication infrastructure in the Cognitive and Neural Systems Department of Boston University.

Evaluation: Demonstration, in unconstrained "street environment", of real-time miniaturized hardware system, with algorithmic implementation of 30 frame/second performance on face-recognition, navigation, target acquisition, and pattern recognition tasks.

10 Robotic Navigation under Visual Guidance (Investigators: P. Gaudiano, A. Harner, E. Sahin)

This research focuses on the development of autonomous mobile robots. Two different modules for different aspects of visually guided navigation will be developed.

Objectives: The long-range goal is to develop an integrated system for control and navigation of autonomous mobile robots in unknown, unstructured environments. Specifically, the goal is to develop mobile robots that can learn about their own dynamics, kinematics, and sensory apparatus so as to operate without supervision in an environment that is unknown or that can undergo unexpected changes. The development of such a system would have direct applications for battlefield scenarios, where autonomous robots could operate under adverse conditions on tasks such as intelligence gathering, mine detection, and disabling of enemy equipment.

Research Questions: Current state-of-the-art robotics are not yet capable of robust, adaptive behavior in unknown or unstructured environments such as a battlefield. Teleoperation is only possible under certain circumstances, and it still suffers from the inability of he teleoperated vehicle to adapt to changes that might result for instance from damages components or unreliable communications channels. An ideal autonomous robot might be able to operate under teleoperation when possible, but otherwise continue to function even if teleoperation becomes impractical.

The development of robust, adaptive robots is a research area that encompasses many problems, such as low-level control, visual recognition, localization, and path planning. Algorithms have been developed in each of these sub-areas, but little exists in terms of robust, integrated systems that combine the results from most of, or all of these research areas.

This project will avail itself of the interdisciplinary expertise that exists in the Department of Cognitive and Neural Systems. The focus will be on unsupervised, autonomous modules for low-level control, visual processing, sensor fusion, and navigation. These will be modified and enhanced as needed to function on real robots.

Evaluation: The models will be tested on real mobile robots. As is well known in the robotics community, implementing any model on a real robot is much more challenging than doing a computer simulation. The effects of realistic sensor noise, unreliable communications and sudden changes in the environment will be evaluated.

A criterion of success is that the competence arises in an unsupervised and self-organizing fashion, so that it can adapt to circumstances that were not foreseen by the algorithm's designer. To demonstrate robustness, a variety of mobile robot platforms are being used (see our Technical Report), and success will be confirmed if the algorithms work on all of them.

11 Neuromorphic VLSI for Battle Awareness (Investigators: A.G. Andreou, G. Cauwenberghs, A. Hubbard)

Objectives: Our work focuses on circuit implementations (VLSI) of boundary and surface completion networks (BCS/FCS) that enhance noisy images. We will develop chips and/or electronic subsystems that can eventually be put on chips. In the case of SAR, we seek to speed computations that would ordinarily be carried out using computers. In such a case, the objective would not necessarily be to conserve power, but to increase speed. In the case of small-scale sensors on a battlefield, a low-power approach needs to be taken. Moreover, for cost purposes, the relative effectiveness of partial BCS/FCS and ART learning systems that are hand-held or involve telemetry can be of significant utility.

Research Questions: The following are key scientific research and engineering issues that will be addressed:

1. Interchip communication: will be a thrust research area in the next year as this is a key technology component for integrating larger systems. The neuromorphic VLSI chips for the BCS model will be refined to overcome present deficiencies and we will aim towards fabricating large designs (at least 128 × 128 cells) per chip. Work will also continue on the address event representation and the mapping of the BCS/FCS architecture on to this emerging communication/computation technology.

2. Synapse design: for the ART family of learning machines will be revisited and we will investigate the use of floating gate structures and improved dynamic analog storage techniques (ADRAM) for multiple level storage. The issue of whether multiple level analog storage is the way of the future is an issue of hot debate in the FLASH ROM community.

3. Resistive grid computation: is still an issue that will be investigated aiming at compact, low power, high speed designs of local computations.

4. Technology limitations: will ultimately determine the performance of future generation systems. We will seek to develop a quantitative framework to predict how the technology imposes constraints on possible implementations of the sophisticated algorithms that are proposed today. This will be done hierarchically at the device, circuit and architecture levels.

Evaluation: The work on VLSI architecture and design will be evaluated in the following ways, that include metrics intrinsic to the VLSI design and architecture work, and metrics that assess the relevance of our work to the program objectives.

The prime evaluation criterion is a performance comparison between the actual hardware modules and the software simulations. In collaboration with the group at Boston University we will run a battery of tests using the SAR data available that have been used for algorithm development. Other relevant criteria are listed below:

1. Computational throughput metrics: Traditional measures of operations per second (OPS) or channel capacity (spatial and temporal) will be employed. In comparisons with other systems, these will be normalized to actual computing costs which are **power**, **size** and **weight**.

2. Design process metrics: The work will be evaluated based on generated portable libraries for characterized circuit designs that can be readily shared with government contractors. Behavioral models for circuits as well as simulators for architectural optimization will also be provided along with the actual layouts. Thus a particular manufacturer could "plug in" specific models for their manufacturing process, resimulate the chip behavior, amend as appropriate, and begin production.

3. Battlefield awareness relevance: The success of this endeavor must ultimately related to a concrete technology transfer effort. We will deploy prototyped systems in actual real world applications and problems of interest to DOD. We are collaborating with colleagues at the Johns Hopkins Applied Physics Laboratory to deploy BCS/FCS VLSI designs that are currently developed under the MURI project on the Flare Genesis autonomous balloon-born sun observatory:
`hurlbut.jhuapl.edu/FlareGenesis/mission.html`.

12 References

Bradski, G. and Grossberg, S. (1995). Fast learning VIEWNET architectures for recognizing 3-D objects from multiple 2-D views. *Neural Networks*, **8**, 1053–1080.

Carpenter, G.A., Grossberg, S., and Lesher, G.W. (1997). The What-and-Where filter: A spatial mapping neural network for object recognition and image understanding. *Computer Vision and Image Understanding*, in press.

Grossberg, S. *et al.* (1997). Automated vision and sensing systems at Boston University. In **Proceedings of the DARPA Image Understanding Workshop**, New Orleans, May 1997. San Francisco: Morton Kaufmann Publishers, in press.

Multidiciplinary University Research Initiative (MURI) Technical Papers

Model-Based Matching by Linear Combinations of Prototypes

Michael J. Jones
Center for Biological and
Computational Learning
MIT, Cambridge, MA 02139
email: mjones@ai.mit.edu

Tomaso Poggio
Center for Biological and
Computational Learning
MIT, Cambridge, MA 02139
email: tp@ai.mit.edu

Abstract

We describe a flexible model similar to (Vetter and Poggio, 1993, 1995 and Jones and Poggio, 1995) for representing images of objects of a certain class, known a priori, such as faces, and introduce a new algorithm for matching it to a novel image and thereby perform *image analysis*. The flexible model is learned from example images (called prototypes) of objects of a class. In this paper we introduce an effective stochastic gradient descent algorithm that automatically matches a model to a novel image by finding the parameters that minimize the error between the image generated by the model and the novel image. Our approach can provide novel solutions to several vision tasks, including the computation of image correspondence, object verification, image synthesis and image compression.

1 Introduction

An important problem in computer vision is to model classes of objects in order to perform *image analysis* by matching the models to views of new objects of the same class, thereby parametrizing the novel image in terms of a known model. Many approaches have been proposed. Several of them represent objects using 3D models, represented in different ways (for a review, see [Besl and Jain, 1985]). Such models are typically quite sophisticated, difficult to build and hard to use for many applications – image matching in particular. A rather different approach is suggested by recent results in the science of biological vision.

There is now convincing psychophysical and even physiological evidence suggesting that the human visual system often uses strategies that have the flavor of object representations based on 2D rather than 3D models ([Edelman and Bulthoff, 1990]; [Sinha, 1995]; [Bulthoff *et al.*, 1995]; [Pauls *et al.*, 1996]). With this motivation, we have explored an approach in which object models are learned from several prototypical 2D images.

Though the idea of synthesizing a model for a class of objects, such as faces or cars, is quite attractive, it is far from clear how the model should be represented and how it can be matched to novel images. In other papers we have introduced a *flexible* model of an object class as a linear combination of example images, represented appropriately. Image representation is the key issue here because in order for a linear combination of images to make sense, they must be represented in terms of elements of a vector space. In particular, adding

[0] This paper describes research done at the Artificial Intelligence Laboratory and within the Center for Biological and Computational Learning at the Massachusetts Institute of Technology. This research is sponsored by grants from ARPA-ONR under contract N00014-92-J-1879 and from ONR under contract N00014-93-1-0385 and by a grant from the National Science Foundation under contract ASC-9217041 and by a MURI grant N00014-95-1-0600. Additional support is provided by ATR Audio and Visual Perception Research Labs, Sumitomo Metal Industries, Kodak, Daimler-Benz and Siemens AG. Tomaso Poggio is supported by the Uncas and Helen Whitaker Chair at MIT's Whitaker College.

two image representations together must yield another element of the space. It turns out that a possible solution is to set the example images in pixelwise correspondence. In our approach, the correspondences between a reference image and the other example images are obtained in a preprocessing phase. Once the correspondences are computed, an image can be represented as a shape vector and a texture vector. The shape vector specifies how the 2D shape of the example differs from the reference image. Analogously, the texture vector specifies how the texture differs from the reference texture (here we are using the term "texture" to mean simply the pixel intensities of the image). The flexible model for an object class is thus a linear combination of the example shapes and textures which, for given values of the parameters, can be rendered into an image. The flexible model is thus a generative model which can *synthesize* new images of objects of the same linear class. But how can we use it to *analyze* novel images? In this paper we provide an answer in terms of a novel algorithm for matching the flexible model to a novel image.

The paper is organized as follows. First, we discuss related work. In section 3 we explain in detail our model. Section 4 describes the matching algorithm. Section 5 shows example models of object classes and presents experiments on matching novel images. Section 6 concludes with a summary and discussion.

2 Related Work

The "linear class" idea of [Poggio and Vetter, 1992] and [Vetter and Poggio, 1995] together with the image representation, based on pixelwise correspondence, used by [Beymer *et al.*, 1993] (see also [Beymer and Poggio, 1996]) is the main motivation for our work. Poggio and Vetter introduced the idea of linear combinations of views to define and model *classes* of objects. They were inspired in turn by the results of [Ullman and Basri, 1991] and [Shashua, 1992] who showed that linear combinations of three views of a single object may be used to obtain any other views of the object (barring self-occlusion and assuming orthographic projection). Poggio and Vetter defined a linear object class as a set of 2D views of objects which cluster in a small linear subspace of $\mathcal{R}^{2n}$ where n is the number of feature points on each object. They showed that in the case of linear object classes rigid transformations can be learned exactly from a small set of examples. Jones and Poggio [Jones and Poggio, 1995] sketched a novel approach to match linear models to novel images that can be used for several visual *analysis* tasks, including recognition. In this paper we develop the approach in detail and show its performance not only on line drawings but also on gray-level images.

Recently we have become aware of several papers dealing with various forms of the idea of linear combination of prototypical images. Choi *et. al.* (1991) [Choi *et al.*, 1991] were perhaps the first (see also [Poggio and Brunelli, 1992]) to suggest a model which represented face images with separate shape and texture components, using a 3D model to provide correspondences between example face images. In his study of illumination invariant recognition techniques, Hallinan [Hallinan, 1995] describes deformable models of a similar general flavor. The work of Taylor and coworkers ([Cootes and Taylor, 1992]; [Cootes and Taylor, 1994]; [Cootes *et al.*, 1992]; [Hill *et al.*, 1992]) on active shape models is probably the closest to ours. It is based on the idea of linear combinations of prototypes to model non-rigid transformations within classes of objects. However, they use a very sparse set of corresponding points in their model (we use dense pixelwise correspondences), and they handle texture differently from us. Our image representation, relying on shape and texture vectors obtained through pixelwise correspondence, seems to be a significant extension which allows us to incorporate texture as well as shape seamlessly into a single framework.

3 Modeling Classes of Objects

The work of Ullman and Basri [Ullman and Basri, 1991] and Poggio and Vetter [Poggio and Vetter, 1992] was based on a representation of images as vectors of the x, y positions of a small number of labeled features - and left open the question of how to find them reliably and accurately. Starting with [Beymer *et al.*, 1993], we have attempted to avoid the computation of sparse features while keeping as much as possible the representation of Ullman and Basri which has – at least under some conditions – the algebraic structure of a vector space. For images considered as bitmaps, on the other hand, basic vector space operations like addition and linear combination are not meaningful. We have argued therefore that a better way to represent images is to associate with each image a shape vector and a texture vector (see for a review [Poggio and Beymer, 1996]).

The *shape vector* of an example image associates to each pixel in the reference image the coordinates of the corresponding point in the example image. The *texture vector* contains for each pixel in the reference image the color or gray level value for the corresponding pixel in the example image. We refer to the operations which associate the shape and texture vectors to an image as *vectorizing the image.* Instead of two separate vectors we can also consider the full texture-shape vector which has dimensionality $N + 2N$ where N is the number of pixels in the image. The term shape vector refers to the 2D shape (not 3D!) relative to the reference image. Note that edge or contour-based approaches are a special case of this framework. If the images used are edge maps or line drawings then the shape vector may include only entries for points along an edge without the need of an explicit texture vector.

The shape and texture vectors form separate linear vector spaces with specific properties. The shape vectors resulting from different orthographic views of a single 3D object (in which features are always visible) constitute a linear vector subspace of very low dimensionality spanned by just two views ([Ullman and Basri, 1991]; see also [Poggio, 1990]). For a fixed viewpoint a specific class of objects with a similar 3D structure, such as faces, seems to induce a texture vector space of relatively low dimensionality as shown indirectly by the results of [Kirby and Sirovich, 1990] and more directly by [Lanitis *et al.*, 1995]. Using pixelwise correspondence [Vetter and Poggio, 1995] and [Beymer and Poggio, 1995] showed that a good approximation of a new face image can be obtained with as few as 50 base faces, suggesting a low dimensionality for both the shape and the texture spaces. As reviewed by [Poggio and Beymer, 1996] correspondence and the resulting vector structure underlie many of the recent view-based approaches to recognition and detection either implicitly or explicitly.

Certain special object classes (such as cuboids and symmetric objects) can be proved to be exactly linear classes (see [Poggio and Vetter, 1992]). Later in the paper we will show that there are classes of objects the images of which – for similar view angle and imaging parameters – can be represented satisfactorily as a linear combination of a relatively small number of prototype images.

3.1 Formal specification of the model

In this section we will formally specify our model which we refer to as a *linear object class model.* An image I is viewed as a mapping

$$I : \mathcal{R}^2 \rightarrow \mathcal{I}$$

such that $I(x, y)$ is the intensity value of point (x, y) in the image. $\mathcal{I} = [0, a]$ is the range of possible gray level values. For eight bit images, $a = 255$. Here we are only considering gray level images, although color images could also be handled in a straightforward manner. To define a model, a set of example images called prototypes are given. We denote these prototypes as $I_0, I_1, \ldots, I_N$. Let I_0 be the reference image. The pixelwise correspondences between I_0 and each example image are denoted by a mapping

$$S_j : \mathcal{R}^2 \rightarrow \mathcal{R}^2$$

which maps the points of I_0 onto I_j, i.e. $S_j(x, y) = (\hat{x}, \hat{y})$ where $(\hat{x}, \hat{y})$ is the point in I_j which corresponds to (x, y) in I_0. We refer to S_j as a *correspondence field* and interchangeably as the shape vector for the vectorized I_j. We define

$$T_j(x, y) = I_j \circ S_j(x, y) = I_j(S_j(x, y)). \quad (1)$$

T_j is the warping of image I_j onto the reference image I_0. In other words, T_j is the set of shape-free prototype images – shape free in the sense that their shape is the same as the shape of the reference image. The idea of the model is to combine linearly the textures of the prototypes all warped to the shape of the reference image and therefore in correspondence with each other. The resulting texture vector can then be warped to any of the shapes defined by the linear combination of prototypical shapes.

More formally, the flexible model is defined as the set of images I^{model}, parameterized by $\mathbf{b} = [b_0, b_1, \ldots, b_N], \mathbf{c} = [c_0, c_1, \ldots, c_N]$ such that

$$I^{model} \circ (\sum_{i=0}^{N} c_i S_i) = \sum_{j=0}^{N} b_j T_j. \qquad (2)$$

The summation $\sum_{i=0}^{N} c_i S_i$ describes the shape of every model image as a linear combination of the prototype shapes. Similarly, the summation $\sum_{j=0}^{N} b_j T_j$ describes the texture of every model image as a linear combination of the prototype textures. Note that the coefficients for the shape and texture parts of the model are independent.

In order to allow the model to handle translations, rotations, scaling and shearing, a global affine transformation is also added. The equation for the model images can now be written

$$I^{model} \circ (A \circ \sum_{i=0}^{N} c_i S_i) = \sum_{j=0}^{N} b_j T_j \qquad (3)$$

where $A : \mathcal{R}^2 \to \mathcal{R}^2$ is an affine transformation

$$A(x, y) = (p_0 x + p_1 y + p_2,\ p_3 x + p_4 y + p_5). \quad (4)$$

The constraint $\sum_{i=0}^{N} c_i = 1$ is imposed to avoid redundancy in the parameters since the affine parameters allow for changes in scale.

When used as a generative model, for given values of $\mathbf{c}$, $\mathbf{b}$ and $\mathbf{p}$, the model image is rendered by computing I^{model} in equation 3. For analysis the goal is, given a novel image I^{novel}, to find parameter values that generate a model image as similar as possible to I^{novel}. The next section describes how.

4 Matching the Model

The analysis problem is the problem of matching the flexible model to a novel image. The general strategy is to define an error between the novel image and the current guess for the closest model image. We then try to minimize this error with respect to the linear coefficients $\mathbf{c}$ and $\mathbf{b}$ and the affine parameters $\mathbf{p}$. Following this strategy, we define the sum of squared differences error

$$E(\mathbf{c}, \mathbf{b}, \mathbf{p}) = \frac{1}{2} \sum_{x,y} [I^{novel}(x, y) - I^{model}(x, y)]^2 \qquad (5)$$

where the sum is over all pixels (x, y) in the images, I^{novel} is the novel gray level image being matched and I^{model} is the current guess for the model gray level image. Equation 3 suggests to compute I^{model} working in the coordinate system of the reference image. To do this we simply apply the shape transformation (given estimated values for $\mathbf{c}$ and $\mathbf{p}$) to I^{novel} and compare it to the shape-free model, that is

$$E(\mathbf{c}, \mathbf{b}, \mathbf{p}) = \qquad (6)$$
$$\frac{1}{2} \sum_{x,y} [I^{novel} \circ (A \circ \sum_{i=0}^{N} c_i S_i) - \sum_{j=0}^{N} b_j T_j(x, y)]^2.$$

Minimizing this error yields the model image which best fits the novel image with respect to the L_2 norm. We use here the L_2 norm but other norms may also be appropriate (e.g. robust statistics).

In order to minimize the error function any standard minimization algorithm could be used. We have chosen to use the stochastic gradient descent algorithm [Viola, 1995] because it is fast and can avoid remaining trapped in local minima.

The summation in equation 7 is over all pixels in the model image. The idea of stochastic gradient descent is to randomly sample a small set of pixels from the image and only compute the gradient at those pixels. This gives an estimate for the true gradient [Viola, 1995]. In our experiments we typically choose only 40 points per iteration of the stochastic gradient descent.

This results in a large speedup over minimization methods – such as conjugate gradient – which compute the full gradient over the whole image.

Stochastic gradient descent requires the derivative of the error with respect to each parameter. These derivatives can be calculated straightforwardly and are given in [Jones and Poggio, 1996].

Our matching algorithm uses a coarse-to-fine approach to improve robustness ([Burt and Adelson, 1983]; [Burt, 1984]) by creating a pyramid of images with each level of the pyramid containing an image that is one fourth the size of the one below. The correspondence fields must also be subsampled, and all x and y coordinates must be divided by two at each level. The optimization algorithm is first used to fit the model parameters starting at the coarsest level. The resulting parameter values are then used as the starting point at the next level. The translational affine parameters (p_2 and p_5) are multiplied by 2 as they are passed down the pyramid to account for the increased size of the images. Section 5 shows a number of examples to illustrate the performance of the matching algorithm.

Another useful technique implemented in our model is principal components analysis. The eigenvectors for both the shape space and texture space are computed independently. The shape space and texture space representations can then be effectively "compressed" by using only the first few eigenvectors (with largest eigenvalues) in the linear combinations of the model (both for shape and texture). We emphasize that the technique performs well without using eigenvectors, which however can provide additional computational efficiency.

5 Examples and Results

The model described in the previous sections was tested on two different classes of objects. The first was the class of frontal views of human faces. A face database from David Beymer, formerly of the MIT AI Lab [Beymer, 1996], was used to create a model of the class of faces. (For a second face model see [Jones and Poggio, 1996].) The Beymer face database – consisting of 62 faces – had been set into correspondence by manually specifying a number of corresponding points and then interpolating to get a dense correspondence field. The second example object class was a set of side views of cars which used 40 car images. The correspondences for this class were also computed by manually specifying a number of corresponding points and then interpolating.

The matching algorithm as described in section 4 was run with the following parameters. The number of samples randomly chosen by the stochastic gradient descent algorithm per iteration was 40 and it was run for 8000 iterations per pyramid level. Three levels were used in the image pyramids.

The running time of the matching algorithm for the Beymer faces using 61 prototypes (with no compression) was about 9 minutes. For the cars, the running time was about 5 minutes using all 40 prototypes (with no compression).

5.1 Faces

The face model was built from the Beymer face database, shown in figure 1. These images were 226 pixels high by 184 pixels wide. The face in the upper left corner was used as the reference face. The correspondences from the reference face to each of the other faces was given by manually specifying a small number of corresponding points. These faces are difficult to put in correspondence due to the hair, beards and mustaches. The results of testing the model's ability to match novel faces are shown in figure 2. Because we only had 62 faces to work with, 61 were used in the model and one was left out so that it could be used as a novel image. Since the hair is not modelled well, the faces in figure 2 have been cropped appropriately. One can see that the matching algorithm produced good matches to the novel faces.

5.2 Cars

As another example of a linear object class model, we chose side views of cars. The car images are 96 pixels high by 256 pixels wide. Forty examples of side views of cars were used to build

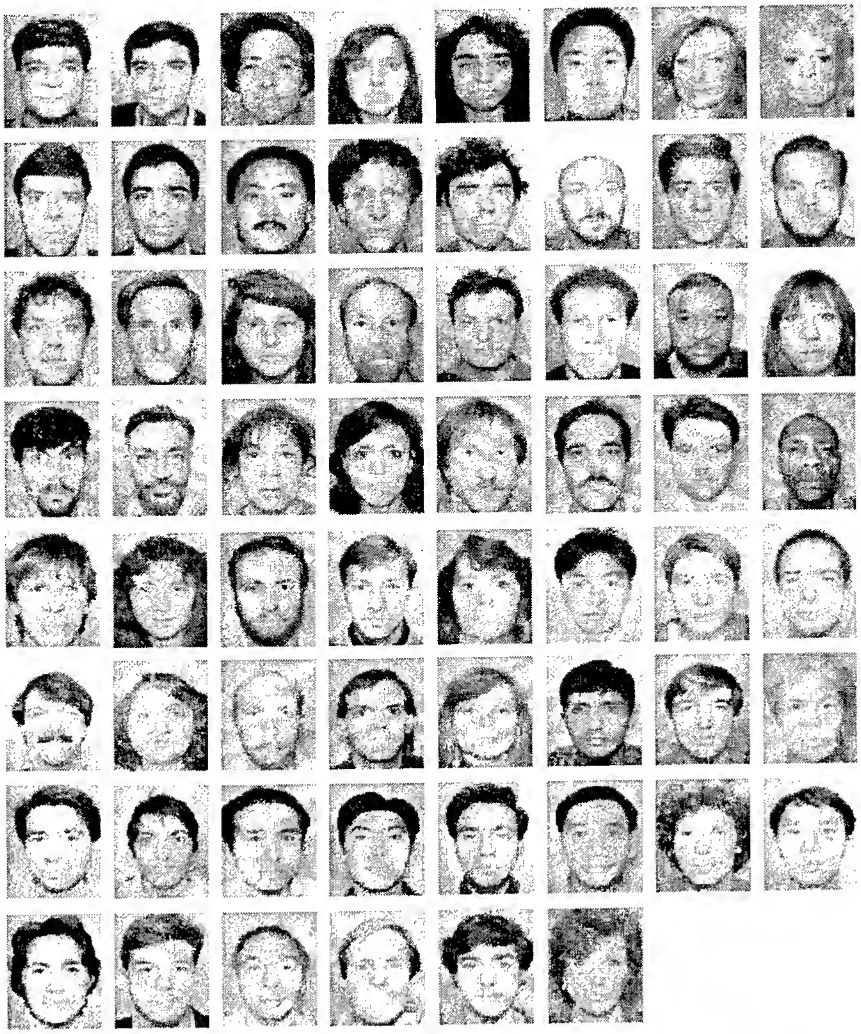


Figure 1: *Database of 62 prototype faces used to create the model of human faces.*

the model. The example images were similar to the novel car images shown in figure 3. The model defined by these prototypes and their correspondences was tested on its ability to match a number of novel cars. Figure 3 shows some example matches to novel cars. The novel cars are matched reasonably well, although not as sharply as the faces. This is probably due to the fact that the correspondences given for the car prototypes are not as accurate as in the case for faces. The main reason for this is the variability of appearance between cars: some cars have features that do not appear on other cars (such as spoilers). Defining correspondences for such features is ambiguous.

6 Conclusions

In this paper we have described a flexible model to represent images of a class of objects and in particular how to use it to analyze new images and represent them in terms of the model. Our flexible model does not need to be handcrafted but can be directly "learned" from a small set of images of prototypical objects. The key idea underlying the flexible model is a representation of images that relies on the computation of dense correspondence between images. In this representation, the set of images is endowed with the algebraic structure of a linear vector space.

The main contribution of this paper is to solve the analysis problem: how to apply the flexi-

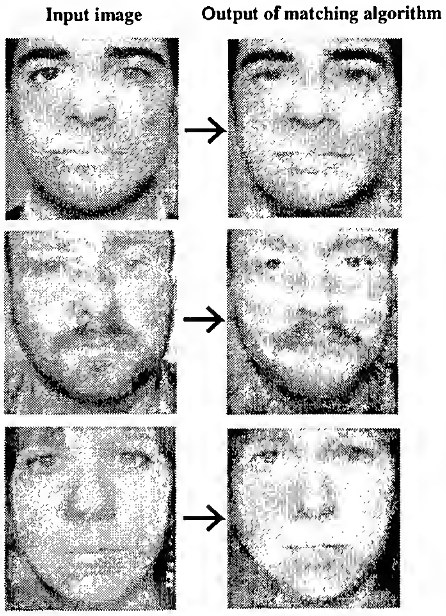


Figure 2: *Four examples of matching the face model to a novel face image.*

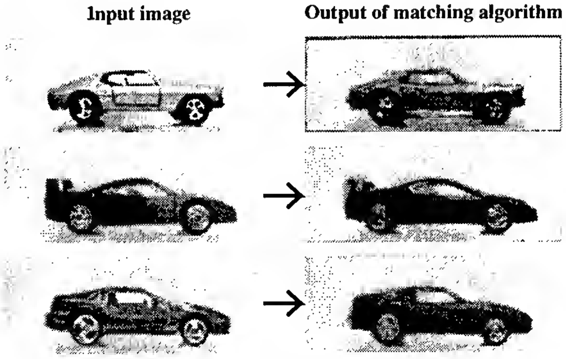


Figure 3: *Three examples of matching the car model to a novel car image, which was not among the model prototypes.*

ble model, so far used as a generative model, for image analysis. Key to the analysis step is matching and a new matching algorithm is the main focus of this paper. The matching algorithm has been shown to be robust to changes in position, rotation and scale of the novel input image. It can also match partially occluded input images [Jones and Poggio, 1996]. We have also described in more detail than in any previous paper the model itself and how to obtain it and learn it from protoype images through pixelwise correspondence. Analysis coupled with synthesis offers a number of significant new applications of the flexible model, including recognition, image compression, correspondence and learning of visual tasks in a top-down way, specific to object classes (e.g. estimation of contours, shape and color). These applications are discussed in more detail in [Jones and Poggio, 1996].

7 Acknowledgements

The authors would like to thank Thomas Vetter, Amnon Shashua, Federico Girosi, Paul Viola and Tony Ezzat for helpful comments and discussions about this work. Michael Oren suggested the notation of section 3.1 in addition to many other very useful comments.

References

[Besl and Jain, 1985]
Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, 1985.

[Beymer and Poggio, 1995] David Beymer and Tomaso Poggio. Face recognition from one example view. A.I. Memo 1536, MIT, 1995.

[Beymer and Poggio, 1996] David Beymer and Tomaso Poggio. Image representations for visual learning. *Science*, 272:1905–1909, June 1996.

[Beymer *et al.*, 1993] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. A.I. Memo 1431, MIT, 1993.

[Beymer, 1996] David Beymer. *Pose-Invariant Face Recognition Using Real and Virtual*

Views. PhD thesis, Massachussetts Institute of Technology, 1996.

[Bulthoff *et al.*, 1995] H. H. Bulthoff, S. Y. Edelman, and M. J. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5(3):247–260, 1995.

[Burt and Adelson, 1983] Peter J. Burt and Edward J. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, 1983.

[Burt, 1984] Peter J. Burt. The pyramid as a structure for efficient computation. In *Multi-Resolution Image Processing and Analysis*, pages 6–37. Springer-Verlag, 1984.

[Choi *et al.*, 1991] Chang Seok Choi, Toru Okazaki, Hiroshi Harashima, and Tsuyoshi Takebe. A system of analyzing and synthesizing facial images. *IEEE*, pages 2665–2668, 1991.

[Cootes and Taylor, 1992] T.F. Cootes and C.J. Taylor. Active shape models - 'smart snakes'. *British Machine Vision Conference*, pages 266–275, 1992.

[Cootes and Taylor, 1994] T.F. Cootes and C.J. Taylor. Using grey-level models to improve active shape model search. *International Conference on Pattern Recognition*, pages 63–67, 1994.

[Cootes *et al.*, 1992] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Training models of shape from sets of examples. *British Machine Vision Conference*, pages 9–18, 1992.

[Cootes *et al.*, 1993] T.F. Cootes, C.J. Taylor, A. Lanitis, D.H. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. In *ICCV*, pages 242–246, Berlin, May 1993.

[Edelman and Bulthoff, 1990] Shimon Edelman and Heinrich Bulthoff. Viewpoint-specific representations in three dimensional object recognition. A.I. Memo 1239, MIT, 1990.

[Hallinan, 1995] Peter Winthrop Hallinan. *A Deformable Model for the Recognition of Human Faces Under Arbitrary Illumination*. PhD thesis, Harvard University, 1995.

[Hill *et al.*, 1992] A. Hill, T.F. Cootes, and C.J. Taylor. A generic system for image interpretation using flexible templates. *British Machine Vision Conference*, pages 276–285, 1992.

[Jones and Poggio, 1995] Michael Jones and Tomaso Poggio. Model-based matching of line drawings by linear combinations of prototypes. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 531–536, 1995.

[Jones and Poggio, 1996] Michael Jones and Tomaso Poggio. Model-based matching by linear combinations of prototypes. A.I. Memo 1583, MIT, 1996.

[Kirby and Sirovich, 1990] M. Kirby and L. Sirovich. The application of the karhunen-loeve procedure for the characterization of human faces. *IEEE*, 12(1):103–108, January 1990.

[Lanitis *et al.*, 1995] A. Lanitis, C.J. Taylor, and T.F. Cootes. A unified approach to coding and interpreting face images. In *ICCV*, pages 368–373, Cambridge, MA, June 1995.

[Pauls *et al.*, 1996] J. Pauls, E. Bricolo, and N.K. Logothetis. Physiological evidence for viewer centered representation in the monkey. In S. Nayar and T. Poggio, editors, *Early Visual Learning*. Oxford University Press, 1996.

[Poggio and Beymer, 1996] Tomaso Poggio and David Beymer. Learning to see. *IEEE Spectrum*, pages 60–69, 1996.

[Poggio and Brunelli, 1992] Tomaso Poggio and Roberto Brunelli. A novel approach to graphics. A.I. Memo 1354, MIT, 1992.

[Poggio and Vetter, 1992] Tomaso Poggio and Thomas Vetter. Recognition and structure from one 2d model view: Observations on prototypes, object classes and symmetries. A.I. Memo 1347, MIT, 1992.

[Poggio, 1990] Tomaso Poggio. A theory of how the brain might work. A.I. Memo 1253, MIT, 1990.

[Shashua, 1992] Amnon Shashua. Projective structure from two uncalibrated images: Structure from motion and recognition. A.I. Memo 1363, MIT, 1992.

[Sinha, 1995] P. Sinha. *Perceiving and recognizing 3D forms.* PhD thesis, Massachussetts Institute of Technology, 1995.

[Turk and Pentland, 1991] M.A. Turk and A.P Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[Ullman and Basri, 1991]
S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992–1006, 1991.

[Vetter and Poggio, 1995] Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. A.I. Memo 1531, MIT, 1995.

[Vetter *et al.*, 1996] Thomas Vetter, Michael Jones, and Tomaso Poggio. A bootstrapping algorithm for learning linearized models of object classes. *submitted*, 1996.

[Viola, 1995] Paul Viola. Alignment by maximization of mutual information. MIT A.I. Technical Report 1548, MIT, 1995.

Orientation Behavior Using Registered Topographic Maps

C. Ferrell
Massachusetts Institute of Technology Cambridge MA

Abstract

The ability to orient toward visual, auditory, or tactile stimuli is an important skill for systems intended to interact with and explore their environment. In the brain of mammalian vertebrates, the Superior Colliculus is specialized for integrating multi-modal sensory information, and for using this information to orient the animal to the source sensory stimuli, such as noisy, moving objects. Within the Superior Colliculus, this ability appears to be implemented using layers of registered, multi-modal, topographic maps. Inspired by the structure, function, and plasticity of the Superior Colliculus, we are in the process of implementing multi-modal orientation behaviors on our humanoid robot using registered topographic maps.

1 Introduction

The ability to orient to sensory stimuli is an important skill for autonomous agents that operate in complex, dynamic environments. In animals, orientation behavior serves to direct the the animal's eyes, ears, nose, and other sensory organs to the source of sensory stimulation. By doing so, the animal is poised to assess and explore the nature of the stimulus with complementary sensory systems, which in turn affects and guides ensuing behavior. Hence, orientation behavior is performed frequently and repeatedly by agents that are tightly coupled with their environment, where perception guides action and behavior assists in more effective perception.

Our approach to implementing orientation behavior on Cog, our humanoid robot [4], is heavily inspired by relevant work in neuroscience [10], [2]. In the brain of mammalian vertebrates, the Superior Colliculus is an organ specialized for producing orientation behavior. In non-mammalian vertebrates (birds, amphibians, etc.), the optic tectum is the analogous organ. The structure of the Superior Colliculus is characterized by layers of topographically organized maps. Collectively, they represent the sensorimotor space of the animal in ego-centered coordinates. These maps are interconnected and interact is such a way that the animal performs orientation movements in response to sensory stimuli.

Topographically organized maps have been discovered throughout the brain of mammalian vertebrates. In addition to the Superior Colliculus, they have been identified in various perceptual areas of the neocortex (the visual, auditory and somatosensory corticies, for instance). It is widely recognized that the organization of these maps are plastic and can be shaped through experience. Subsequently, cortical maps have garnered a lot of attention, and a variety of work has explored the phenomena of self-organizing feature maps, [6], [8], [7].

The rest of this paper is organized as follows. First we will briefly cover the organization, structure, and function of the Superior Colliculus, as our implementation is strongly inspired by what is understood about this organ. Next we present the state of our implementation at the time this paper was written, as well as extensions currently under development. Finally, we present tests and results of our system to date, and conclude with a brief description of ongoing work and future directions.

2 The Superior Colliculus

The Superior Colliculus is a midbrain structure composed of seven laminar layers. The deep layers are those believed to play a role in orientation behavior. An important function of the Superior Colliculus is to pool sensory inputs from different modalities and redirect the corresponding sensory organs (eyes, ears, nose) to fixate on the source of the signal.

[0] Support for this research was provided by a MURI grant under the Office of Naval Research contract N00014–95–1–0600.

2.1 Organization of the Superior Colliculus

Localized regions of the Superior Colliculus consist of neurons with receptive fields that form topologically organized maps. Each map corresponds to either a single modality or a combination of modalities. In the cat, there are visuotopic maps representing motion in visual space, somatotopic maps yielding a body representation of tactile inputs, and spatiotopic maps of auditory space encoding inter-aural time differences (ITD) and inter-aural intensity differences. Hence, a sensory stimulus originating from a given direction will elicit activity in the corresponding region of the appropriate sensory map. There are also motor movement maps consisting of pre-motor neurons whose movement fields are topologically organized. In the cat, these exist for the eyes, head, neck, body, ears.

2.2 The Role of Map Registration

These multi-modal maps overlap and are aligned with each other so that they share a common multisensory spatial coordinate system. The maps are said to be *registered* with one another when this is the case. Arranging multi-modal information into a common representational framework within each map and aligning them allows the information to interact and influence each other. There are several advantages to this organizational strategy. First, it is an economical way of specifying the location of peripheral stimuli, and for organizing and activating the motor program required to orient towards it; thereby allowing any sensory modality to orient the other sensory organs to the source of stimulation. Second, it supports enhancement of simultaneous sensory cues. Stimuli that occur in the same place at the same time are likely to be interrelated by common causality. For instance, a bird rustling in the bushes will provide both visual motion and auditory cues. During enhancement, certain combinations of meaningful stimulus become more salient because their neuronal responses are spatio-temporally related. Once the multi-modal maps are aligned, neuronal enhancement (or depression) is a function of the temporal and spatial relationships of neural activity among the maps.

2.3 Development and Experience Dependent Plasticity

During development, the organization of the topographic maps as well as the registration between different maps is plastic. For each map, its representation of space is use dependent. A number of people have modeled the phenomena of self organizing feature maps (SOFMs) using neural networks [6], [1], [5]. Plasticity of map registration has been studied in the inferior and superior colliculus of young barnyard owls, where the registration of the auditory map to the visual map shifts according to experience [2], [3].

3 A Developmental Approach to Orientation Behavior

Registered topographic maps form a substrate upon which multi-modal information can be integrated to produce coherent behavior. How are these topographic maps formed? How do they become registered with one another? How is the organization of the ensemble guided by experience?

3.1 The Framework

In our framework, a map is a two dimensional array of elements where each element corresponds to a site in the map. The maps are arranged into interconnected layers, where a given map can be interfaced to more than one map. Each connection is uni-directional, so recurrent connections between maps require both a feedforward connection and a feedback connection. The activity level of sites on one map is passed to another map thorough these connections, hence the input to a given map is a function of the spatio-temporal activity of the maps feeding into it and the connectivity between these maps. Currently, all connections have equal weights, although this could change in the future. The output of a given map is its spatio-temporal activity pattern. What this pattern of activity represents depends upon the map: if it is a visuotopic map, it could represent motion coming from a particular direction in the visual field; if it is an oculomotor map, it could encode a motor command to move the eyes, and so forth.

The smallest map ensemble capable of producing an observable behavior consists of a sensory input map, a motor output map, and an established set of connections between them. The input map could have a fairly rigid structure consisting simply of time-differenced intensity images. Because visual information already contains a spatial component, this simple map is topographic without any additional tuning. The motor map could also be fixed where a given site on the map corresponds to a given motor command. If the motor commands vary linearly with motor space, for instance, this map is also topographically organized. Assuming the cameras are motionless, a moving object occupies a localized region in the visual field, and correspondingly causes a localized intensity difference (an active region) in the time-differenced image map. If there exists connections from this region of the time-difference map to the appropriate region of the oculomotor map, then a motion stimulus in the visual field activates the corresponding region of the time-difference map, which in turn excites the connected region of the oculomotor map, which evokes the necessary camera motion to foveate the stimuli.

3.2 Developmental Mechanisms

Plasticity can be introduced into the simple system above in two ways: 1) the map organization could change so that a given map site could correspond to different locations in space. 2) The connections between maps could change so that a given site could change which site(s) it connects to on the other map.

In animals, as described in section 2.3, the organization of the maps and the registration between maps is tuned during the critical period of development. Several mechanisms and models have been proposed to account for this organizational process. The mechanisms we use for map organization and alignment on Cog are inspired by similar mechanisms [6]. However, different combinations of mechanisms are used depending on what is being learned: i.e. tuning the organization within a map, registering different sensory maps, or registering sensory maps and motor maps.

A variety of mechanisms determine how map connections are established. Guided by sensori-motor experience, these mechanisms govern how connections are modified to improve behavioral performance.

- *Competition*: There is competition between concurrently active sites where only the most active site is modified per trial. In our system, the most active is currently approximated as the centroid of activity of the active region. Furthermore, each site of a given map can only form a limited number of connections to the other map. So, candidate map sites compete to determine those that can connect to a given site on the other map.

- *Locality, neighborhood influences*: The neighboring sites around the most active site are also updated each trial. The amount a neighboring site is adjusted decays with distance from the maximally active site. This mechanism penalizes long connections and encourages topographic organization. The size of the neighborhood can vary over time. Typically, it starts off fairly large until the map displays some rough topographic organization, then it decreases as the map undergoes fine tuning adjustments.

- *Error correction*: It is not sufficient that the maps are topographically organized and aligned – they must be organized and interfaced so that the agent performs well in its environment. For tight feedback loop sensorimotor tasks (such as saccading to a visual stimulus), an appropriate error signal is very important and useful for tuning the behavior of the system. Naturally, the error signal must be a good measure of performance and obtainable at a fast enough rate to enable on-line learning. Connections are modified to reduce the discrepancy between current performance and desired performance. The magnitude of the correction is proportional to the size of the error on that trial.

- *Correlated temporal activity*: Hebbian mechanisms are often used for self-organizing processes. By strengthening connections between simultaneously active sites, they are useful for relating information between different sensory maps.

- *Learning rate*: The magnitude of the adjustment for each trial is also proportional to the learning rate. The learning rate can vary over time, where it starts of relatively large for course tuning, and then decreases for finer adjustments.

3.3 A Simple Behavior

In this section we look at an example to see how these mechanisms are applied to forming and organizing these multiple maps to perform a task. A simple orientation task is the ability to saccade to noisy, moving stimuli (clapping hands, shaking a rattle, etc.). We say that a good saccade centers the stimulus in the fovea camera's field of view, whether the stimulus is seen in the wide field of view or the fovea field of view. We assume that the system favors information from the foveal view because it is of higher resolution than the peripheral view and thereby can be used to perform a more accurate saccade.

Experience dependent plasticity could play a role in several ways. It could be used to guide the representational organization of the auditory and visual motion maps, guide the registration between the auditory and visual motion maps, or guide the registration of the the sensory maps to the oculomotor map.

In this paper, we concentrate on the registration of sensory maps to motor maps.

Each mapping process can be viewed as learning a multi-modal map that registers the information from two other modality maps. We call the multi-modal map the *registration map*, and the other maps could be either sensory maps, motor maps, or both. One of the modality maps provides the rough spatial organization of the multi-modal map. We call this map the *receptive field map*. Typically it has a topographic representation of space. Often a retinotopic map is used, for instance. The second modality map, which may or may not be topographic, is registered to the first map through the multi-modal map. Note that each site of a modality map connects to only one site on the registration map, but the same site on the registration map could connect to mulitple sites on the modality maps.

3.3.1 Registration of Sensory-Motor Maps

An example of aligning sensor and motor maps is registering the oculomotor map with the visual motion map. In this case, the receptive field map is the visual motion map (in retinotopic coordinates), the registration map is a visuo-motor map (also in retinotopic coordinates), and the third map is the oculomotor map (in eye motor coordinates). Regions in the motor map correspond to motor movements that could foveate a stimulus. Initially the visual map and the oculomotor map are connected to the registration map with broad, overlapping receptive fields. When the motion map is stimulated and the site of maximal activity is determined (typically the centroid of the stimulated region), the corresponding region of the oculomotor map is stimulated. The site of maximal response of the motor map is taken as the motor command, and the corresponding motor movement is evoked. This movement orients the eye to the stimulus. Once oriented, the motion stimulus stimulates a different region in the visual motion map. The visual error is computed as the difference from centroid of motion to the center of the field of view. This error is used to update the connections responsible for the orientation movement to reduce the error in the future. Hence, the primary developmental mechanisms are competition, neighborhood updates, and error correction.

4 Architectural Organization

To date, the sensory-motor map registration task has been implemented on Cog's hardware and is shown in figure 1. The diagram shows how the processes are arranged on Cog's MIMD computer. Currently, five processing nodes are used:

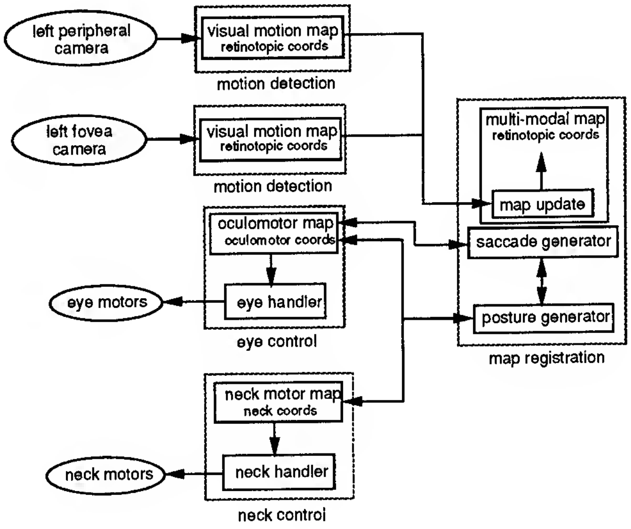


Figure 1: This diagram shows how the multi-modal topographic maps are arranged on Cog's computational hardware. Currently five processors are used: two visual processors, two motor control processors, and one processor which performs the developmental mechanisms. See text for further explanation.

- *Peripheral motion processor*: Contains the peripheral visual motion map. It computes the difference between consecutive left peripheral camera images at 15 frames/s. It also determines the most active site (the centroid of motion) and a visual error signal.

- *Fovea motion processor*: Contains the fovea visual motion map. It computes the difference between consecutive left fovea camera images at 15 frames/s. It also determines the most active site (the centroid of motion) and a visual error signal.

- *Registration processor*: Contains the visuomotor map and carries out the developmental process. It receives motion information from the vision processor and determines which motion information to use. If fovea motion is present, it ignores the information from the peripheral camera. It also translates the most active site on the visual map to the region of activity on the motor map, and passes this information to the oculomotor processor. After the motion is performed, it uses the error signal from the vision processors to update the registration map connections according to developmental mechanisms.

- *Oculomotor processor*: Contains the oculomotor map. Upon receiving the site of activ-

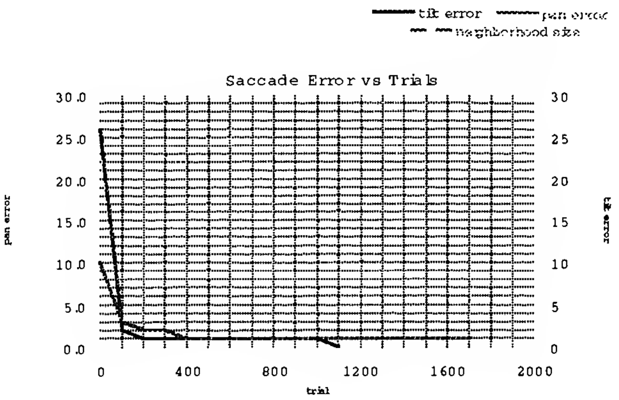


Figure 2: Registration data for aligning the visual motion map and the oculomotor map. The data is derived from the registration map, converting sites in visuotopic coordinates to activated sites in the motor movement map (pan and tilt) required to foveate the stimulus. Each trial, the neighborhood size is set equal to the larger value of the average error measures (pan or tilt). Initially the maping is random, with a neighborhood size of 26 and a learning rate of .25. By $trial = 100$, the average error is reduced to $(3°, 2°)$, and by $trial = 400$ it is reduced to $(1°, 1°)$. After training, it converges to an average error $\leq (1°, 1°)$.

ity, it commands the motors to perform the movement. It also sends an "efferent copy" to the registration processor, so the registration processor can ignore visual motion information while the cameras are moving.

- *Neckmotor processor*: Contains the neckmotor map. It is commanded by the registration processor to move the neck around so the motion stimulus is seen from many different places in the visual field. Currently, the neck is primarily used for the training processes. However, soon it will incorporated into the orientation behavior.

5 Tests

To date we have run experiments to test whether the implementation we have described learns the registration between the retinotopic visual motion map and the oculomotor map. A sampling of our results are shown in figure 2.

So far, experiments have been performed using the left eye only. The system will be extended to handle both eyes when stereopsis and vergence capabilities are implemented. Motion information from both eyes will be fused and used to excite the visuotopic motion map. Conflicts between the eyes will be resolved during this fusion stage. The simplest approach would be to resolve conflicts via a dominant eye mechanism. Another method could involve exciting the visuotopic map with the stonger of the two excitations coming from each eye. These two methods along with other possibilities need to be explored. Most likely, a combination of methods will be implemented.

To learn the registration between the peripheral motion map with the oculomotor map, we trained Cog over a number of trials while it looked at a continuously moving stimulus. At the beginning of each trial, the robot changes its posture (centers its eyes and moves its neck to a random location). This places the motion stimulus in a different location in its visual field. Currently Cog explores the center $20° \times 20°$ of the peripheral visual field, which corresponds to a 20×20 region of the registration map. The robot uses the visual information to stimulate the oculomotor map and perform the saccade. The visual error is then acquired, and the registration between the maps is updated according to the rule:

$$\Delta m(x,y) = \rho \times \epsilon(x,y) \times N(x,y) \tag{1}$$

where:

- $m(x,y)$ is the value of site (x,y) of registration map m. Recall that this value represents the connection from the visual motion map site to the corresponding oculomotor map site. The learning process involves updating these intermap connections.
- (x^*,y^*) is the site of maximal activity of the motion map. For this application, it corresponds to the site of maximal activity of the registration map as well.
- ρ is the learning rate.
- $\epsilon(x,y) = target(x,y) - m(x,y)$. It is an error distance measure between the motion map site $m(x,y)$ and the target site. This measurement is made after the saccade motion finishes. Note that $target(x,y)$ is the center of the field of view for the saccade learning task.
- r is the neighborhood radius.
- $N(x,y) = f(1 - \frac{|(x^*,y^*)-(x,y)|}{r})$. It is the neighborhood update function that decays linearly with distance from the site of maximal activity (x^*,y^*). Threshold function, f, sets the result equal to zero if its argument is negative. So, for site locations outside radius r of (x^*,y^*) , $N(x^*,y^*) = 0$.

6 Continued Work

Currently, we are extending these tests to include the full visual field, and continuing our experiments with dynamically varying neighborhood size and learning rate parameters. Soon we will explore the self organizing properties of the representation of visual information by implementing a SOFM for visual motion. We will also begin efforts to register the visual motion with an auditory ITD map, as well as investigate the dynamics of development when self-organization and registration mechanisms are run simultaneously. We expect to see evidence for different developmental time scales as the robot learns the orientation task.

With the above work in place, we will extend the system to include the neck and body degrees of freedom so that the robot can perform full body orientation behavior. This will complicate the current task by adding additional degrees of freedom that must complement each other. We will continue to investigate the issue of developmental time scales since more complicated behaviors will have to develop incrementally and bootstrap off of existing behaviors. We would like to integrate the full orientation behavior with reaching and manipulation tasks currently under parallel development by other members of the group [9].

7 Conclusions

This paper describes an implementation of orientation behavior on Cog using registered topographic maps. We have presented biological evidence that this is an effective method for orienting to multimodal stimuli in animals. We have also presented a series of mechanisms and methods for developing this behavior on Cog over time. This biologically inspired framework gives us the opportunity to explore several interesting issues. It allows us to investigate using dynamic spatio-temporal representations of sensory-motor space to integrate multi-modal information and produce a unified behavior. It also allows us to investigate the dynamics of development using mechanisms of experience dependent plasticity. Ongoing work is promising.

Acknowledgements

A number of people have made this work and the Cog Project possible. Among those who have contributed to the ongoing development of Cog are (in alphabetical order): Mike Binnard, Rod Brooks, Robert Irie, Eleni Kapogannis, Matt Marjanovic, Yoky Matsuoka, Brian Scasselatti, Nick Shectman, Rene Schaad, and Matt Williamson.

References

[1] H. Bauer and K. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, 1992.

[2] M. Brainard and E. Knudsen. Experience-dependent plasticity in the inferior colliculus: A site for visual calibration of the neural representation of auditory space in the barn owl. *The Journal of Neuroscience*, 13(11):4589–4608, 1993.

[3] M. Brainard and E. Knudsen. Dynamics of visually guided auditory plasticity in the optic tectum of the barn owl. *Journal of Neurophysiology*, 73(2):595–614, 1995.

[4] R. Brooks and L. A. Stein. Building brains for bodies. *Autonomous Robots*, 1:1:7–25, 1994.

[5] R. Durbin and G. Mitchison. A dimension reduction framework for understanding cortical maps. *Nature*, 343(6259):644–647, 1990.

[6] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[7] K. Obermayer, H. Ritter, , and K. Schulten. A principle for the formation of the spatial structure of cortical feature maps. *Proceedings of the National Academy of Science USA*, 87:8345–8349, 1990.

[8] H. Ritter and K. Schulten. Kohonen's self-organizing maps: Exploring their computational capabilites. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 109–116, San Diego, CA, 1988.

[9] B. Scassellati, M. Williamson, and M. Marjanovic. Self-taught visually-guided pointing for a humanoid robot. In *Proceedings of the 4th Intl. Conference on Simulation of Adatpive Behavior*, Cape Cod, MA, 1996.

[10] B. Stein and M. Meredith. *The Merging of the Senses.* A Bradford Book, Cambridge, MA, 1993.

A Bootstrapping Algorithm for Learning Linear Models of Object Classes

Thomas Vetter
Max-Plank-Institut
für biologische Kybernetik
72076 Tübingen, Germany
email: vetter@mpik-tueb.mpg.de

Michael J. Jones
Center for Biological and
Computational Learning
MIT, Cambridge, MA 02139
email: mjones@ai.mit.edu

Tomaso Poggio
Center for Biological and
Computational Learning
MIT, Cambridge, MA 02139
email: tp@ai.mit.edu

Abstract

Flexible models of object classes, based on linear combinations of prototypical images, are capable of matching novel images of the same class and have been shown to be a powerful tool to solve several fundamental vision tasks such as recognition, synthesis and correspondence. The key problem in creating a specific flexible model is the computation of pixelwise correspondence between the prototypes, a task done until now in a semiautomatic way. In this paper we describe an algorithm that automatically bootstraps the correspondence between the prototypes. The algorithm – which can be used for 2D images as well as for 3D models – is shown to synthesize successfully a flexible model of frontal face images and a flexible model of handwritten digits.

1 Introduction

In recent papers we have introduced a new type of flexible model for images of objects of a certain class. The idea is to represent images of a certain type – for instance images of frontal faces – as the linear combination of prototype images and their affine deformations. This flexible model can be used as a generative model to *synthesize* novel images of the same class. It can also be used to *analyze* novel images by estimating the model parameters via an optimization procedure. Once estimated the model can be used for indexing, for recognition, for image compression and for image correspondence.

At the very heart of our flexible models is an image representation in terms of which a linear combination of images makes sense. For a set of images to behave as vectors, they must be in pixelwise correspondence (see [Poggio and Beymer, 1996]). Our model uses pixelwise correspondence between example images and should not be confused with techniques which use linear combinations of *images* such as the so-called eigenfaces technique ([Kirby and Sirovich, 1990]; [Turk and Pentland, 1991]). In our approach, the correspondences between a reference image and the other example images are obtained in a preprocessing phase. Once the correspondences are computed, an image is represented as a *shape vector* and a *texture vector.* The shape vector specifies how the 2D shape of the example differs from a reference image and corresponds to the flow field between the two images. Analogously, the texture vector specifies how the texture differs from the reference texture. Here we are using the term "texture" to mean simply the pixel intensities (grey level or color values) of the image. Our flexible model for an object class is then a linear combination

[0] This research is sponsored by grants from ARPA-ONR under contract N00014-92-J-1879 and from ONR under contract N00014-93-1-0385 and by a grant from the National Science Foundation under contract ASC-9217041 (this award includes funds from ARPA provided under the HPCC program) and by a MURI grant N00014-95-1-0600. Additional support is provided by ATR Audio and Visual Perception Research Laboratories, Sumitomo Metal Industries, Kodak, Daimler-Benz and Siemens AG.

of the example shape and texture vectors.

1.1 A key problem: creating the model from prototypes

The distinguishing aspect of our flexible models is that they are linear combinations of prototype shape and texture vectors and not of images ([Beymer and Poggio, 1996]). The prototypical images must be vectorized first, that is correspondence must be computed among them.

This is a key step and in general a difficult one. It needs to be done only once at the stage of developing the model. At run-time no further correspondence is needed – and in fact the model can be used to compute correspondence if necessary. In our past papers we computed correspondence between the prototypes with automatic techniques such as optical flow. Sometimes, however, we were forced to use interactive techniques requiring the user to specify at least some of the correspondences (see for instance [Lines, 1996]). An automatic technique that could set prototypes in correspondence would be therefore desirable even if very slow. In addition, any claim of biological plausibility would require the demonstration of such a technique.

In this paper we describe a bootstrapping technique that seems capable of computing correspondence between prototypical images in cases in which standard optical flow algorithms fail.

2 Linear models

For the formal specification of the model, we refer the reader to [Jones and Poggio, 1996]. The relevant notation used in that paper and refered to later in this one is as follows. Images I_0 through I_N are the prototype images. The flow field $S_j : \mathcal{R}^2 \rightarrow \mathcal{R}^2$ maps the points of I_0 onto I_j. S_j is also called the shape vector. $T_j = I_j \circ S_j$ is the texture vector for image I_j.

2.1 Matching the model

The matching algorithm is also described in detail in [Jones and Poggio, 1996]. In brief, the matching algorithm attempts to minimize the error between the novel input image and the current guess for the best fitting model image. This is done by a gradient descent algorithm which adjusts the parameters of the model to minimize the L2 error between the novel and model images.

2.2 Optical Flow

For some prototypes, the pixelwise correspondences from the reference image to the prototype can be found accurately by an optical flow algorithm. We have mostly used the multiresolution, laplacian-based, optical flow algorithm described in [Bergen and Hingorani, 1990].

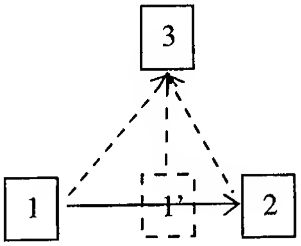


Figure 1: Given the flexible model provided by the combination of image 1 and image 2 (in correspondence), the goal is to find the correspondence between image 1 (or image 2) and the novel image 3. Our solution is to first find the linear combination of image 1 and image 2 that is closest to image 3 (this is image 1') and then find the correspondences from image 1' to image 3 using optical flow. The two flow fields can then be composed to yield the desired flow from image 1 to image 3.

3 Bootstrapping the synthesis of a flexible model

Suppose that we have a flexible model consisting of N protoypes in correspondence. It is tempting to try to use it to compute the correspondence to a novel image of an object of the same class so that it can be added to the set of prototypes. The obvious flaw in this strategy is that if the flexible model can compute good correspondence to the new image then there is no need to add it to the flexible model since it will not increase its expressive power. If it can't, then the new prototype cannot be incorporated as such. A possible way out of this conundrum is to bootstrap the flexible model by using it together with an optical flow algorithm.

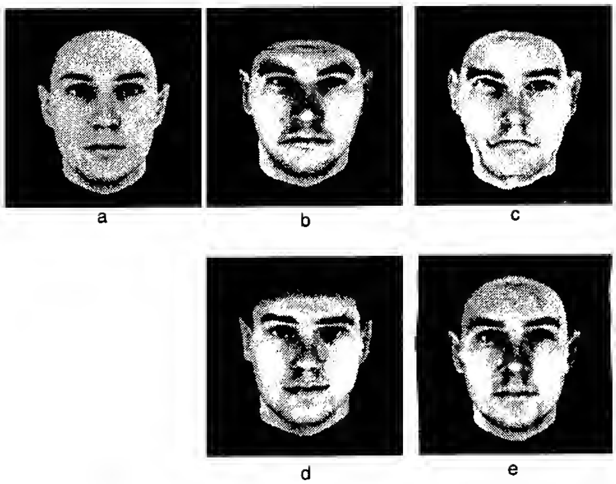


Figure 2: This figure shows the basic idea behind bootstrapping. Image (a) is the reference face. Image (b) is a prototype. Image (c) is the image resulting from backward warping the prototype onto the reference face using the correspondences found by an optical flow algorithm. Image (d) is the model image which best matches the prototype using a model consisting of 20 prototypical faces (which did not include image (b)). Image (e) is the image resulting from backward warping the prototype onto the reference face using the flow field which was composed from matching the face model and then running an optical flow algorithm between image (d) and image (b) to further improve the correspondences. This is the basic step of the bootstrapping algorithm.

3.1 The basic recursive step: improving the match with optical flow

Suppose that an existing flexible model is not powerful enough to match a new image and thereby find correspondence with it. The idea is first to find rough correspondences to the novel image using the (inadequate) flexible model and then to improve these correspondences by using an optical flow algorithm. This idea is illustrated in figure 1. In the figure, a model consisting of image 1 and image 2 (and the correspondences between them) is first fit to image 3. Call the best fitting linear combination of images 1 and 2 image 1'. The correspondences are then improved by running an optical flow algorithm between the intermediate image 1' and image 3. Notice that this technique can be regarded as a class specific regularization of optical flow.

3.1.1 Example

An example of our basic step is shown in figure 2. In this figure, an optical flow algorithm is used to find the correspondences from image (a) to image (b). The resulting correspondences are not very good as shown by image (c) which is the backward warp of image (b) according to the correspondences found by optical flow. Image (c) should have the texture of image (b) and the shape of image (a). A better way to find the correspondences is to first fit a model of faces to image (b), by using as a model 20 prototype face images (with known correspondences). The model is matched to image (b) as described in section 2.1. The resulting best match is shown as image (d). Next, optical flow was run between image (d) and image (b) to further improve the correspondences found by the matching algorithm. The two correspondence fields are combined to get the correspondences from image (a) to image (b). Image (e) is the backward warp of image (b) according to the final correspondence. Comparing image (c) with (e) shows better correspondences are found with bootstrapping.

3.2 The bootstrapping algorithm

The idea of bootstrapping is to start from a small flexible model consisting of just two prototypical images and to increase its size (and representation power) by iterating the recursive step described above, progressively adding new images by setting them in correspondence with the model.

There are two main problems with building a linear flexible model. The first one is to choose the reference image, relative to which shape and texture vectors are represented. The second is to automatically compute the correspondences even in cases in which optical flow fails.

In principle, any example image could be used as the reference image. However, small peculiarities in an image can influence the matching process. Thus, an image which is close to all images is more reliable, since the computation of the correspondence is more stable for small distortions than for bigger ones. The average image of the whole data set, for which the average distance to the whole data set is by definition at minimum, is the optimal reference image. Since the correspondences between the images cannot be computed correctly in one step, the average has to be computed in an iterative procedure. Starting from an arbitrary image as the preliminary reference, a (noisy) correspondence between all other images and this reference is first computed using an optical flow algorithm. On the basis of these correspondences an average image can be computed, which now serves as a new reference image. This procedure of computing the correspondences and calculating a new average image is repeated until a stable average (vectorized) image is obtained.

The correspondence fields obtained through the optical flow algorithm from this final average image to all the examples are usually far from perfect. The bootstrapping idea is to improve the correspondences by applying iteratively the basic step described above while also increasing the expressive power of the flexible model. We could incorporate into the flexible model one new image at each timestep. Instead, we have implemented an equivalent algorithm in which the first step is to form a linear object model from the correspondences obtained from all images with optical flow. Since some of these correspondence fields are not correct and all are noisy, this algorithm uses only the most significant fields as provided by a standard PCA decomposition of the shape and the texture vectors. Instead of adding new images, the algorithm increases with successive iterations the number of principal components, ordered according to the associated eigenvalues (the allowed range of parameters of the selected principal components can also be increased with a similar effect). At each iteration a flexible model is selected and used to match each image. The optical flow algorithm estimates correspondence between the image and the approximation provided by the flexible model. This field is then added to the correspondence field implied by the matched model, giving a new correspondence field between the reference image and the example. The correspondence fields, obtained by this procedure, will finally lead to a new average image and also to new principal components which can be incorporated in an improved flexible model. Iterating this procedure with increasing expressive power of the model (by increasing the number of principal components) leads to stable correspondence fields between the reference image and the examples. The number of iterations as well as the increasing complexity of the model can be regarded as regularization parameters of this bootstrapping process.

3.2.1 Pseudo code of an efficient algorithm

1A: Selecting a reference image.

```
Select some image $I_i$ as reference image $I_{ref}$.
**Until** convergence do {
    **For** all $I_i$ {
        Compute correspondence field $S_i$
        between $I_{ref}$ and $I_i$ using optical flow.
        Backwards warp $I_i$ onto $I_{ref}$ using $S_i$
        to get the texture map $T_i$.
    end **For**}
    Compute average over all $S_i$ and $T_i$
    Forward warp $T_{avg}$ using $S_{avg}$ to
        create $I_{average}$
    Convergence test: is $I_{avg} - I_{ref} < limit$ ?
    Copy $I_{avg}$ to $I_{ref}$;
end **Until** }
```

1B: Computing the correspondence.

```
Until number $n$ of principal components used
in the linear model is maximal {
    Perform a principal component analysis on
        $\{S_i\}$ and separately on $\{T_i\}$.
    Select the first $n$ principal components for
        the linear model.
    Approximate each $I_i$ by the linear model
        with $I_i^{model}$.
    Compute correspondence field $S_i'$ between
        $I_i^{model}$ and $I_i$ using optical flow
    Combine $S_i'$ and $S_i^{model}$ to $S_i^{new}$
    Backwards warp $I_i$ onto $I_{ref}$ using $S_i^{new}$
        to get the texture map $T_i$.
    Copy all $S_i^{new}$ to $S_i$.
    Increase number $n$ of principal components
        used in the linear model.
end Until }
```

4 Results

The method described in the previous sections was tested on two different classes of images. One class was frontal views of human faces and the second was handwritten digits. We describe here only the first class (for the second see [Vetter *et al.*, 1997]).

4.1 Face images

4.1.1 Data set

Frontal images of 130 caucasian faces were used in our experiments. The images were originally rendered for psychophysical experiments [Troje and Bülthoff, 1995] under ambient illumination conditions from a data base of three-dimensional human head models recorded with a laser scanner ($Cyberware^{TM}$). All faces were without makeup, accessories, and facial hair. Additionally, the head hair was removed digitally (but with manual editing), via a vertical cut behind the ears. The resolution of the grey-level images was 256-by-256 pixels and 8 bit.

Preprocessing: First the faces were segmented from the background and aligned roughly by automatically adjusting them to their two-dimensional centroid. The centroid was computed by evaluating separately the average of all x, y coordinates of the image pixels related to the face independent of their intensity value.

4.1.2 Evaluation

The method described in the previous sections was successfully applied to all face images available.

The step involving synthesis of the reference (average) image was tested for each image as a starting image in the algorithm. As a convergence criteria we used a theshold on the minimum average change of the pixel gray value (0.3, whereas the range was 256). The threshold was reached in every case within 5 iterations and mostly after 3. The final reference images could not be distinguished under visual inspection. One of these reference images is shown in the second column of figure 3; the same reference image was used for the final correspondence finding procedure.

Optical flow yields the correct correspondence between the reference image and each example image only in 80% of all cases. In the remaining cases the correspondence is partly incorrect, as shown in figure 3. The center column shows the images which result from backward warping the face images (left column) onto the reference image using the correspondence fields obtained through the optical flow algorithm. In the first iteration of the correspondence finding procedure the first 2 principal components of the shape vectors (that is of the correspondence fields) and of the textures vectors are used in the flexible model. Then the correspondence field provided by matching with the flexible model is combined with the correspondence field obtained by the optical flow algorithm between the face image and its flexible model approximation. The backward warps using these correspondence fields are shown in the fourth column. The correspondence fields were iterated by slowly increasing the number of principal components used in the flexible model. After four iterations with 2, 10, 30 and 80 principal components, the correspondence fields between the reference face and all example images did not reveal any obvious errors (right column).

5 Conclusions

The bootstrapping algorithm we described is not a full answer to the problem of computing

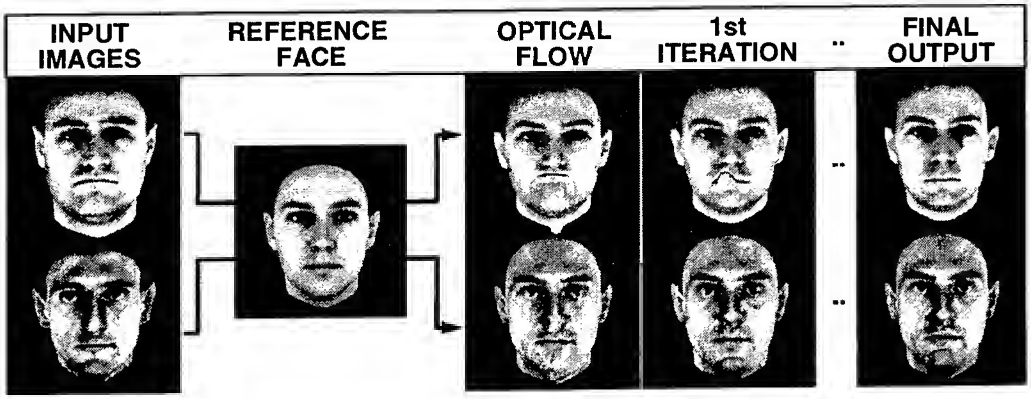


Figure 3: Two of the most difficult faces in our data set. The correspondence between face images (left column) and a reference face can be visualized by backward warping of the face images onto the reference image (three columns on the right). The correspondence obtained through the optical flow algorithm does not allow a correct mapping (center column). The first iteration with a linear flexible model consisting of two principal components already yields a significant improvement (top row). After four iterations with 10, 30 and 80 components, respectively, all correspondences were correct (right column)

correspondence between prototypes. It provides however an initial and promising solution to the very difficult problem of automatic synthesis of the flexible models from a set of prototypical examples. Notice that we have used multiresolution optical flow as one part of our bootstrapping algorithm. In principle other matching techniques could be used within our bootstrapping scheme.

References

[Bergen and Hingorani, 1990] J.R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center, April 1990.

[Beymer and Poggio, 1996] David Beymer and Tomaso Poggio. Image representations for visual learning. *Science*, 272:1905–1909, June 1996.

[Jones and Poggio, 1996] Michael Jones and Tomaso Poggio. Model-based matching by linear combinations of prototypes. A.I. Memo 1583, MIT, 1996.

[Kirby and Sirovich, 1990] M. Kirby and L. Sirovich. The application of the karhunen-loeve procedure for the characterization of human faces. *IEEE*, 12(1):103–108, January 1990.

[Lines, 1996] Steve Lines. The photorealistic synthesis of novel views from example images. Master's thesis, MIT, 1996.

[Poggio and Beymer, 1996] Tomaso Poggio and David Beymer. Learning to see. *IEEE Spectrum*, pages 60–69, 1996.

[Troje and Bülthoff, 1995] N. Troje and H.H. Bülthoff. Face recognition under varying pose: The role of texture and shape. *Vision Research*, 36(12):1761–1771, 1995.

[Turk and Pentland, 1991] M.A. Turk and A.P Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[Vetter *et al.*, 1997] Thomas Vetter, Michael J. Jones, and Tomaso Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition (to appear)*, 1997.

Desktop Programmable Pixel-Parallel Accelerator for High Speed Image Processing

J.C. Gealow, N.S. Love, G. Hall, I. Masaki, C.G. Sodini *
Microsystems Technology Laboratories
Massachusetts Institute of Technology
E-MAIL: jgealow@mtl.mit.edu,nsl@mtl.mit.edu
ghall@mtl.mit.edu, masaki@mtl.mit.edu, sodini@mtl.mit.edu

Abstract

MIT's Modular Vision System architecture consists of three parts: Pentium processor, software modules, and hardware modules. The hardware modules are further classified into application-specific and programmable accelerators. This paper describes a programmable pixel-parallel accelerator. Features include its compact physical size (desktop), efficient control path architecture, convenient programming environment with the C++ programming language, and real-time capability (e.g., 9.9 msec for optical flow). Examples of experimented applications include template matching, optical flow, and stereo matching.

1 Introduction

Typical low-level image processing tasks, including optical flow, template matching, stereo matching, and smooth and segmentation computations, require thousands of operations per pixel for each input image. Traditional general-purpose computers, such as PCs, are not capable of performing such tasks in real time.

The mismatch between the demands of low-level image processing tasks and the characteristics of conventional computers motivates investigation of alternative architectures for a programmable hardware module as part of MIT's Modular Vision System shown in Figure 1. MIT's Modular Vision System is being developed as a common computing platform with high performance, low cost, modular components. It consists of commercial microprocessors with complementary hardware modules. The structure of image processing tasks suggests employing a single instruction stream, multiple data stream (SIMD) design: an array of processing elements, one per pixel, sharing instructions issued by a single controller.

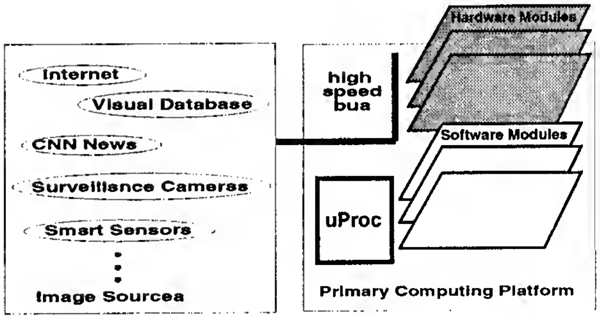


Figure 1: MIT's Modular Vision System Architecture

Massively parallel SIMD supercomputers, providing large processing element arrays, are able to perform image processing tasks in real time. But their million-dollar price tags preclude widespread use. These systems, intended for a broader range of applications, offer capabilities far beyond those necessary for processor-per-pixel image processing. For example, the Connection Machine Models CM-1 and CM-2 [Tucker and Robertson, 1988] provide over four thousand bits of memory per processing element. Sophisticated processing element communication networks employed in the CM-1 and

*This work was partly supported by contract N00014-95-1-0600 from the Office of Naval Research

CM-2 account for a large portion of the cost of the machines, but for most low-level image processing tasks, the networks offer little advantage over much simpler designs.

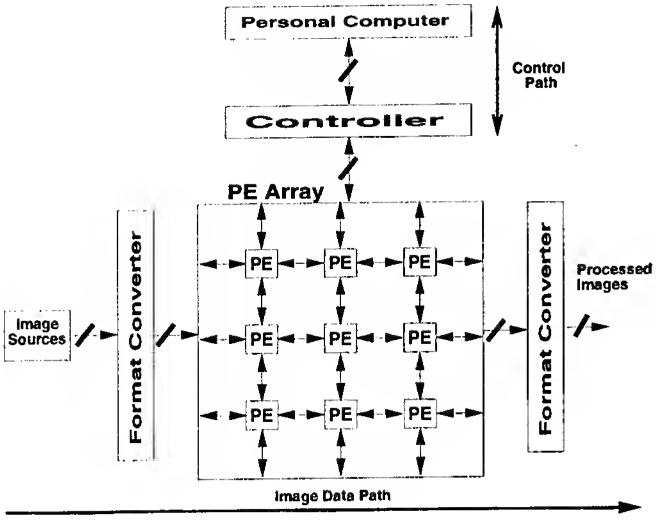


Figure 2: Image processing system using fully integrated processing elements.

We present a system design capable of supporting real-time image processing in a desktop computer environment. Figure 2 illustrates principal components of the system design. Modern VLSI technology permits the economical implementation of large processor-per-pixel arrays. A two-dimensional network connects the processing elements. As shown in the figure, control and data paths are distinct. The processing element array receives instructions from the controller, which is managed by the host computer. With the image data path, data format is converted for mapping image data over the array processor and then re-converted after the processing.

2 System Architecture

System architecture is defined significantly by two factors: the control scheme of array-processor chips with the host computer and the architecture of the array-processor chips. For the control scheme, one might consider using software executed on a host computer to generate low-level instructions in order to minimize system cost. In this approach, low-level instructions are transferred over the host's backplane bus. Unfortunately, there are two serious problems with direct sequential control. First, the host computer may not be able to compute instructions rapidly enough to keep the array busy. Second, the rate at which the host can deliver instructions to the array is limited by bus transaction delays.

To reduce the bus bandwidth and host computation required to sustain array activity, modern SIMD systems typically employ a multi-level control hierarchy. In a conventional two-level design, a microcontroller is placed between the host and the processing element array. The microcontroller executes microcode, interpreting macroinstructions issued by a host and transferred over the host's backplane bus. The microcontroller produces the instructions executed by the array. Typical macroinstructions produce many array instructions. Thus, the demands on the host computer and bus are reduced. The Connection Machine [Hillis, 1985] and the Massively Parallel Processor [Batcher, 1985] employ two-level designs. The associative string processor (ASP) testbed developed by Aspex Microsystems [Habiger and Lea, 1993] and the Vastor processor [Loucks, 1982] employ three-level control hierarchies, adding an additional level of interpretation between the host computer. and the microcontroller.

While the conventional hierarchical control strategy may be suitable for SIMD supercomputers, it is not appropriate for less expensive systems. One drawback is that to generate array instructions at a rate commensurate with the speed of the processing element array, a sophisticated, fast micrôcontroller is needed. The microcontroller design must include one or more functional units to perform the arithmetic and logical operations involved in producing array instructions. Given the amount of computation generally required to decode macroinstructions and produce array instructions, the frequency of the controller clock must be several times that of the array clock.

A second drawback to conventional hierarchical control is the burden of providing appropriate software support. With both a control path and one or more functional units, the microcontroller amounts to a special-purpose computer. Separate programs are required for the host and controller. Getting these two programs to work together can be more than twice as hard as writing a single program. The microcontroller also requires its own set of development tools, including a compiler and a debugger.

Run-time instruction computation is not needed for real-time image processing. The same low-level tasks are repeated for each image. An efficient control strategy exploits this property. Sequences of array instructions are generated by the host computer before processing begins and are stored in the controller. Macroinstructions are reduced to simple calls telling the controller which sequence to send to the processing element array. The controller can be simplified because it does not have to decode and interpret complex macroinstructions.

With our controller architecture, sequences of microinstructions, generated by the host, are held in the control store. Microinstructions include both array instructions and sequencer inputs. To initiate a sequence of microinstructions, the host computer writes the starting address of the sequence into the opcode register. The sequencer steps through the control store, producing one array instruction every clock cycle.

Our control strategy has several important features:

(1) **Simple controller hardware**
Since array instructions are generated by the host computer, the controller need not perform arithmetic and logical operations. Thus, the controller does not include a data path. Instead, array instructions are merged into the control path. Furthermore, the controller need not operate with a higher clock frequency than the processing element array.

(2) **High array utilization**
During processing, the host computer need not issue individual instructions to the processing element array. Thus, provided that sequences are sufficiently long, array utilization is not unacceptably limited by host speed or by bus transaction delays.

(3) **Unified software development**
All system software can be created and compiled on the host computer with existing software development tools.

3 Chip Design

A processing element combines a 128-bit DRAM column with one-bit-wide logic. Three latches provide inputs to a function generator. Eight control signals select between the 256 three-input Boolean functions. Additional latches hold write data and provide a local write-enable signal.

Processing elements are interconnected to create a two-dimensional rectangular array, matching the structure of image data. Input signals from adjacent processing elements are combined with the function generator result according to control signals. With the interconnection network extended across chip boundaries, multiple chips may be used to form processing element arrays matching the size of large images.

The chip was fabricated in a 3.3 ASIC process available through the MOSIS Service. The chip has eight blocks of 512 processing elements. At the center of each block is a twin cell DRAM array with 128 rows and 512 columns. The cell structure is similar to early planar cells with metal bitlines and polysilicon wordlines and platelines [Rideout, 1979], but incorporates a second-level metal wordline shunt. Sense amplifiers are placed at the bottom of the DRAM array.

The layout pitch of processing element logic is double the memory column pitch. Arrays of 256 logic units are placed above and below the DRAM array. There are no column decoders—logic circuits are connected directly to the bitlines. The pitch-matched processor implementation maximizes the bandwidth between memory and logic and minimizes processing element area [Elliot et al., 1979]. Interface circuits use a 3.3V supply to make input and output levels compatible with standard 3.3V parts. Wordline drivers and platelines also use a 3.3V supply. All other circuits use a 2.5V supply. The chip characteristics are listed in Table 1.

Table 1: Chip Characteristics

Channel length	0.6 μm (drawn)
Polysilicon pitch	1.5 μm (w/o contacts)
Metal1 pitch	2.1 μm (contacted)
Metal2 pitch	2.4 μm (contacted)
Metal3 pitch	3.0 μm (contacted)
Proc. elements	4096 (64 × 64)
Memory	512 Kb (twin cell DRAM)
Devices	2.7 M
Pads	144
Die size	9.7 × 8.1 mm^2
Memory cell size	7.2 × 7.2 μm^2
Power supplies	$V_{DD} = 2.5$ V (internal)
	$V_{HH} = 3.3$ V (interface)
	$V_{PP} = 3.3$ V (wordline)
Cycle time	60 ns
Pow. dissipation	300 mW (typical)

4 Programming Environment

Very fine-grained parallel processors with one-bit-wide processing element·logic present challenging software problems. Instruction sets are very primitive: simple parallel arithmetic, comparison, and data movement operations require sequences of many array instructions. In addition, different processing element array implementations employ different memory structures, provide different instructions, and require different computational algorithms. To facilitate application development and maintenance, we created a programming framework supporting our system structure. The framework hides details of array and controller implementations, allowing programmers to focus on application issues.

We developed the programming framework using the C++ programming language [Stroustrup, 1991]. C++ provides facilities for defining new types (classes) that act in the same way as built-in types. In effect, these facilities allow the language to be augmented to support additional concepts. We used this capability, implementing the framework as a library of C++ classes.

A processing element array coupled with a controller is represented by a system class. Systems classes are implemented using an interface to array and controller hardware or using array and controller simulation code. The simulation option provides a valuable tool for architecture development work. An object of class sequence represents a set of controller and array instructions which may be loaded into a region of the control store on the controller and invoked within an application. Application programs employ processing element arrays by directing the execution of sequences by systems.

5 Experimental Results

Edge detection, template matching, optical flow, median filtering, stereo matching, and smoothing and segmentation have been implemented on the system producing real-time results. The underlying parallelism in these applications provide excellent examples to present the processors capabilities. This section details one of the applications and later displays the processing times of the applications tested.

5.1 Template Matching

Template matching consists of comparing a template image to another image to determine if the a portion of the image is the same as the template. The algorithm used determines the matching by computing the difference between the template and the image being matched. The template is normally smaller than the image that is being searched. Every location is tested by calculating the sum of the differences at each location. Figure 3 shows the template and Figure 4 shows the image which was searched. The white square marks where the template has matched the image. The processing time for this template matching operation is $\tilde{6}$.3 ms. The template size which truly determines the number of instructions is 21x31 pixels. The size of the searched image is not relevant in the speed because the operations are performed in parallel. Every pixel of the image is tested with a pixel in the template image at the same time.

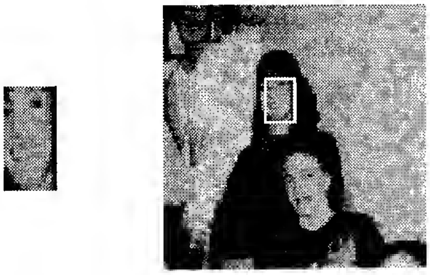


Figure 3: 21x31 Template Image & Test Image

Table 2: Execution times of applications implemented on system.

App.	Spec.	Exec.
Smoothing & Segmentation	200 convolution threshold 20	4.8 ms
Edge Detection	vertical & horizontal edges	277 μs
Template Matching	20x20 template	3.25 ms
5x5 Median Filtering	———————	274 μs
Optical flow	5x5 sup. region 7 pixel max disp.	9.9 ms
Stereo Matching	———————	4.0 ms

5.2 Processing Times

This section contains some processing time results for the applications implemented on the system. The software module can also estimate the processing time of the hardware module. When running the software module a count of array instructions can be obtained which can be used to calculate a conservative approximation of the processing time in the hardware module.

Execution Time = # of instructions x 60 ns + # of sequence executions x 2.5 μs

The 60 ns is the clock speed of the controller and 2.5 μs is the amount of time for the controller and desktop to interface. The execution times are generally less than (1/0.9) x # of instructions x 60 ns. Actual execution time depends on array utilization. Table 2 shows the execution times of various applications implemented on 8 bit images.

6 Future Work

Two efforts are underway. First, the array size is being expanding from 128x128 to 256x256 without increasing the board size. To avoid the increase of the board size, the array processor chips are being repackaged with smaller packages and more dense board are being designed. Second, the host computer is being switched from a workstation to PC for better cost-performance.

References

[Batcher, 1985] Kenneth E. Batcher, "Array control unit," in *The Massively Parallel Processor* (J. L. Potter, ed.), pp. 170-190, Cambridge, MA: The MIT Press, 1985.

[Elliot et al., 1979] Duncan G. Elliott, W. Martin Snelgrove, and Michael Stumm, "Computational RAM: A memory-SIMD hybrid and its application to DSP," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 839-852, June 1979.

[Gealow et al., 1996] Jeffrey C. Gealow, Frederick P. Herrmann, Lawrence T. Hsu, and Charles G. Sodini, "System Design for Pixel-Parallel Image Processing", *IEEE Transactions on VLSI Systems*, vol. 4, no. 1, pp. 32-41, March 1996.

[Gealow and Sodini, 1995] Jeffrey C. Gealow and Charles G. Sodini, "A pixel-parallel image processor using logic pitch-matched to dynamic memory," accepted for presentation at the *Symposium on VLSI Circuits*, June 1997.

[Habiger and Lea, 1993] Claus M. Habiger and R. Mike Lea, "Hybrid-WSI: A massively parallel computing technology?," *Computer*, vol. 26, no. 4, pp. 50-61, April 1993.

[Hillis, 1985] W. Daniels Hillis, *The Connection Machine.* Cambridge, MA: The MIT Press, 1985.

[Loucks, 1982] Wayne M. Loucks, Martin Snelgrove, and Safwat G. Zaky, "A vector processor based on one-bit microprocessors," *IEEE Micro*, vol. 2, no. 1, pp. 53-62, February 1982.

[Rideout, 1979] V. Leo Rideout, "One-device cells for dynamic random-access memories: A tutorial," *IEEE Transactions on Electron Devices*, vol. ED-26, no. 6, pp. 839-852, June 1979.

[Stroustrup, 1991] Bjarne Stroustrup, *The C++ Programming Language.* Reading, MA: Addison-Wesley, second ed., 1991.

[Tucker and Robertson, 1988] Lewis W. Tucker and George G. Robertson, "Architecture and applications of the Connection Machine," *Computer*, vol. 21, no. 8, pp. 26-38, August 1988.

Hardware for Content-Based Image Queries

Aaron Lipman and Woodward Yang*
Division of Engineering and Applied Sciences, Harvard University
Pierce Hall, 29 Oxford Street, Cambridge, MA 02138
E-MAIL: lipman@eecs.harvard.edu,woody@eecs.harvard.edu
HOMEPAGE: http://vlsi.harvard.edu

Abstract

Image databases are finding increased use in multimedia tasks such as pattern recognition and image library searches. A content based approach to image database searching is useful in a variety of tasks such as object recognition, and image understanding. Typically, this involves time-consuming searches through the database to find relevant images. The I/O bandwidth required to quickly hash thorugh a large database can be on the order of gigabits-per-second. We have designed VLSI components for searching databases with high-dimensional data, such as image templates, to find the k closest examples to an input query as determined by a programmable metric using a massively parallel search. This nearest-neighbor approach can be used directly for classification, or in conjunction with any number of algorithms that exploit 'local' information. We have integrated the search hardware on a single die with 16 Mbits of DRAM to perform a highly parallel search with an on-chip memory badwidth of 6.4 Gbits/second. Multiple chips can be connected to form a system capable of over 100 Gbits/second.

1 Introduction

Image databases are exploited in multimedia applications such as pattern recognition and image understanding and classification. For example, consider OCR (optical character recognition). Several examples of handwritten characters can be stored as 16 × 16 templates, or 256-dimensional data vectors. Given a query character template, the retrieval of the best matches to the query from the database can produce a classification match. Other examples include scanning a database to find various objects and featrues, such as faces, or cars.

The nearest-neighbor algorithm can be used to retrive close matches to an input query, as defined by a given distance metric. Operations can be performed on the retreived matches to make a classification decision, or can be used in conjunction with more complex algorithms as a tool for image understanding. Though there are software implementations of nearest neighbor searches utilising tree structures to index quickly into a multidimensional database in logarithmic time in the number of stored data vectors, they have an exponential dependence on the dimensionality of the input data. For highly-dimensional data they essentially perform an exhaustive search of the database. [1] [2] Thus, a highly parallel exhaustive search is the most practical for highly-dimensional data.

Though the nearest-neighbor algorithm requires massive computation, memory, and I/O, a hardware implementation can make it practical for many applications. We have pursued a unified approach, combining memory, distance computation, and sorting on a single chip to perform a highly parallel nearest neighbor search in hardware. The design is flexible, allowing for a wide range of data sizes, and full scalability of memory size, processing speed, and the number of close matches that can be retrieved.

A short review of previous hardware implementations will be given, followed by a description of the chip architecture, and it's prospective performance.

2 Previous Hardware Approaches

An examination some of previous hardware approaches for the nearest neighbor algorithm reveals a few drawbacks. [3].

*This work was supported in part by an NSF Young Investigator Award (MIP-92-57964), a grant for High Performance Computing and Communication from NSF/DARPA (GC-R-334138), and a grant from the ONR Multidisciplinary University Research Initiative (CT-S-608521).

Some chips, such as the Intel Ni1000 [4] and the SGS-Thomson CAM [5] offer storage of a few thousand data vectors on chip, with a fast nearest-neighbor classification result, but are much too restrictive in terms of small fixed data vector sizes, (256 dimensions and 64 dimensions respectively), and poor bit resolution, which limits them to a few special purpose applications. Others, with better bit resolution such as the IBM ZISC036 chip offer only a limited amount of on chip data storage, (36 data vectors for the ZISC036) which is too small to store a reasonably sized database.

There are some embedded-DRAM chips [6] [7] which contain SIMD parallel processors with embedded DRAM, capable of performing nearest-neighbor computations, but they are implemented with only a few hundred bits of DRAM per processor, and offer limited communication between processors, which provides an inferior implementation of communication heavy algorithms like the nearest-neighbor computation.

The high data bandwidth and communication requirments of the nearest neighbor application should beconsidered in the design of hardware, as well as ample storage for large databases, and highly dimensional data.

3 Architecture for k-Nearest Neighbors Retrieval

An exhaustive database search for k-nearest neighbors consists of two basic operations - distance calculation and sorting. Distances between an input query, and each example in a database must be calculated, with the results sorted in a priority queue of size k such that only the k smallest distances are kept. The distance calculations are easily parallelized by assigning a distance calculator to each example in an example database. Once the distances are computed by each processor, they can be sorted in parallel, as new distances are calculated. This basic parallel, pipelined architecture is illustrated in figure 1, which shows the three basic components of the architecture - memory, distance processing, and sorting. The level of parallelism is restricted only by the practical concerns of chip area, and flexibility for example databases of varying size and dimensionality.

Custom hardware was built with a bit-serial design, allowing for a simple interface between memory, processors, and sorters, as well as high-speed operation. Each clock cycle, a single bit is read from the memory into each of the processors. With a large number of memory banks on chip, and clock speeds ranging from 100Mhz-200Mhz, a high memory bandwidth can be achieved.

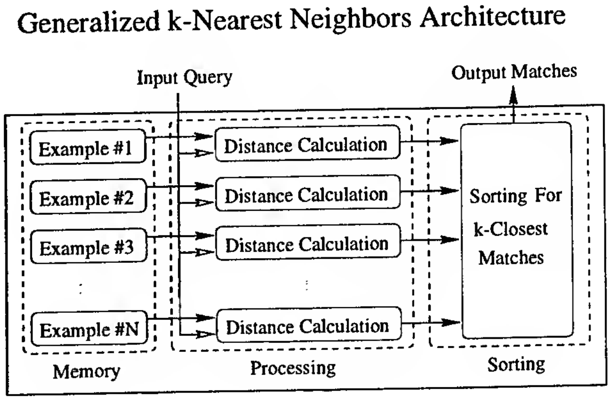


Figure 1: Block diagram of architecture. Examples are stored in memory, and presented in parallel to distance calculation processors along with a common input query. The distances are computed, and sorted in parallel.

4 Hardware Implementation

4.1 Specifications

Target design specifications for the hardware have been selected to be appropriate for a large range of potential applications. The system should be scalable to store a large number of examples, in the range of 10^3 to 10^7 examples. The number of nearest neighbors that can be retrieved should also be scalable. As a compromise between flexibility, and ease of hardware implementation, a programmable weighted L1-norm was chosen as the distance metric:

$$Distance = \sum_{i=1}^{n} \alpha_i |x_i - y_i| \qquad (1)$$

The constants α_i are chosen to emphasize the importance of selected example features or dimensions. The example vectors are encoded using 16 bit signed integers for each dimension, and 8 bit positive unsigned integer weights. The processors will hold 32 bit accumulated distance measurements. Given the example and weight bit resolutions, examples up to 256 dimensions can be accommodated without overflow in the accumulated distance measure.

Examples of larger dimensionality can be accommodated by reducing the bit precision of the weights, or by assuming that examples which produce overflow are too far away from the query to be considered relevant, and can be ignored.

4.2 An Embedded Memory System

The custom distance-calculation and sorting hardware was merged with commercial DRAM. Specifically, a 16-Mbit DRAM organized into 64 256-Kbit blocks, incorporating 64 distance calculation processors, and a 64-entry sorting array. Figure 2 shows a die photo of the chip. Each 256 Kbit block is interfaced with an L1-Norm processor, which is in turn connected to a 64 × 1 array of sorters, acting as a single priority queue slot. The total die area is roughly $17mm \times 17mm$. Though the DRAM is designed in a 0.35 micron process, the unsuitibility of the fabrication process for standard logic requires 1.0 micron design rules for the custom layout. Improvements in the process should allow for significant reductions in die area. The operation of the chip can take place at speeds between 100-200 Mhz, as the DRAM data is slowly sensed in parallel, and then fed to the processors at a high rate in a serial fashion, insuring a new bit per clock cycle for each processor. Runing at 100 Mhz, a chip of 64 processors would have a peak memory-to-processor bandwidth of 6.4 Gbits/second. Multiple chips can be used to scale processor speed, memory storage, and the number of nearest neighbors that can be retrieved.

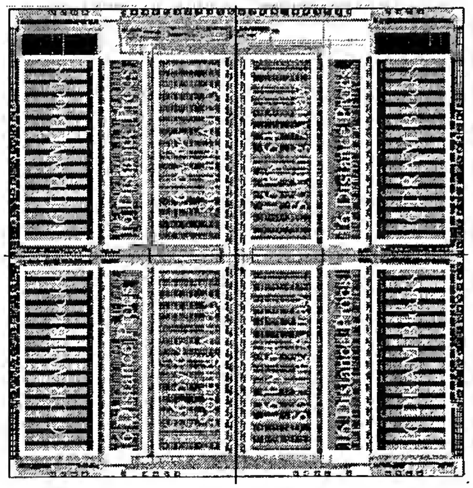


Figure 2: Die photograph of embedded DRAM chip.

5 System Issues

5.1 Projected Performance

As an example of system performance, consider a database of 4,000 16 × 16 templates of written characters. One template requires 4 Kbits of storage (16 bits per dimension * 256 dimensions). 4,000 of these templates requires 16 Mbits of storage, which fits on a single embedded-DRAM chip. An HP-755 workstation takes about 180ms to retrieve the 64 closest matches to an input query. At 100 MHz, a single bit serial L1-norm processor would require only 160ms to retrieve the 64 closest matches (The L1-norm processor loads one bit per clock cycle, and will take 16 Mcycles to process all the examples. At 100 MHz, this takes 160ms, if we ignore pipeline latency.) A single chip with 64 processors would require 2.56ms to retrieve the 64 closest matches, and a system with 16 chips working together would require only $160\mu s$. Thus, speedup factors of greater than 10^3 are feasible using the special-purpose hardware. Other image data, color templates of faces, or medical images for example, can be stored and searched in a similar manner. The massive I/O bandwidth between memory and processing on the special purpose hardware can perform at rates above 100 Gbits/second for a system of 16 chips containing 32 Mbytes of DRAM connected to 1024 processors operating in parallel.

References

[1] J. H. Friedman, J. L. Bentley, R. A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software,* vol. 3, No. 3, pp. 209-226, Sept. 1977.

[2] H. Niemann and R. Goppert, "An Efficient Branch-and-Bound Nearest Neighbour Classifier," *Pattern Recognition Letters,* vol. 7, pp. 67-72, Feb. 1988.

[3] P. Ienne, "Digital Systems for Neural Networks," In R. Kerwin, and P. Papamichalis, editors, *Digital Signal Processing Technology*, vol. CR57 of *Critical Reviews Series*, pp. 314-45. Orlando, FL: SPIE OPtical Engineering, 1995.

[4] C. Park, K. Buckmann, J Diamond, U. Santoni, S. C. The, M. Holler, M. Glier, C. L. Scofield, and L. Nunez, "A Radial Basis Function Neural Network With On-Chip Learning," *Proceedings of the International Joint Conference on Neural Networks,* vol. 3, pp 3035-3038, 1993.

[5] A. Kramer, R. Canegallo, M. Chinosi, D. Doise, G. Gozzini, L. Navoni, P. L. Rolandi, M. Sabatini, "55GCPS CAM Using 5b Analog Flash," *Digest of Technical Papers, IEEE International Solid-State Circuits Conference*, vol. 40, pp 44-45, 1997.

[6] MM32k Preliminary Documentation, Copyright 1995, Current Technology, Inc.

[7] D. G. Elliott, W. M. Snelgrove, & M. Stumm, "Computational Ram: A Memory-SIMD Hybrid and its Application to DSP," *CICC 1992* May 6, session 30.6 (paper 361)

A Coarse-grained, Reconfigurable Image Coprocessor

Alexander Bugeja and Woodward Yang*
Division of Engineering and Applied Sciences, Harvard University
Pierce Hall, 29 Oxford Street, Cambridge, MA 02138
E-MAIL: bugeja@eecs.harvard.edu,woody@eecs.harvard.edu
HOMEPAGE: http://vlsi.harvard.edu

Abstract

We present a coarse-grained, reconfigurable coprocessor for compute and I/O intensive image processing tasks. While FPGA based reconfigurable coprocessors utilize fine-grain, general purpose, homogenous functional blocks, we have implemented a coarse-grain, heterogenous architecture with systolic processor arrays, programmable length line buffers, and specialized processor blocks which is significantly faster and more area efficient. Using 0.6 μm CMOS technology, the 180,000 transistor image coprocessor was implemented in only 4.12mm × 2.59mm, performed 7Gops/sec at 33MHz, and consumed 600mW at 3.3V power supply. The performance of the reconfigurable coprocessor is estimated for a typical computer vision tasks such as block matching, database comparison, and template searching.

1 Introduction

Machine vision tasks are characterized by the extremely demanding computations and I/O bandwidth requirements for even simple real-time visual inspection and analysis tasks. While typical audio processing tasks can be implemented in real-time with DSP chips, the I/O bandwidth, memory capacity, and computation throughput required for image processing tassks are several orders of magnitude beyond the capabilities of even supercomputer performance. In particular, it has been estimated that at least several hundred teraflops would be required to approach the basic capabilities of human visual systems.

Although several orders of magnitude performance improvement over state-of-the-art microprocessors can be realized through the careful design and implemenation of full custom VLSI coprocessors, these special purpose processors are only optimized to perform a single task. In response, reconfigurable coprocessors have been proposed as a possible solution and have been implemented using FPGAs (Field Programmable Gate Arrays) with fine-grained, homogenous, general purpose functional blocks [Knittel et al., 1996]. In contrast, we present a novel coarse-grained, reconfigurable image coprocessor which can be efficiently programmed for high-speed execution of a range image comparison tasks such as block matching, robust template correlation, and pyramidal image searching. This image coprocessor is called the G2 (generation two) here to distinguish it from an earlier design (the G1 [Gilbert and Yang, 1993]) which utilized a fixed systolic processor array.

Block matching (or image correlation [1]) is computationally and I/O intensive and is encountered in several areas of computer vision and image processing such as automated visual inspection systems, face recognition and identification systems, motion estimation, and video compression systems employing motion estimation (e.g. many MPEG and H.261 compliant systems). The very large computational expense of block matching task requires as much as as three quarters of the total processing power in typical video codecs [Fujiwara et al., 1992]. This expense is obviously greatest in the case of full search block matching, as compared to reduced search schemes, some of which are detailed in [Pirsch et al., 1995]. Full search block matching however has the desirable properties of relatively simple control and guaranteed correct results and is an ideal candidate for VLSI implementation due to the inherent

*This work was supported in part by an NSF Young Investigator Award (MIP-92-57964), a grant for High Performance Computing and Communication from NSF/DARPA (GC-R-334138), and a grant from the ONR Multidisciplinary University Research Initiative (CT-S-608521).

[1] This term is equivalent to block matching and will be used interchangeably.

parallelism which can be exploited to compute block matching scores in real-time. The core systolic processor array in the G2 is optimized for the full search block matching task and is utilized as a basic primitive function for performing a variety of other tasks.

2 Architecture

The G2 processor was designed as a high-speed coprocessor for use over fast local buses. In particular, the 180,000 transistor design was fabricated in 0.6 μm CMOS technology on a 4.12mm × 2.59mm die, performed 7 Gop/s at 33 MHz (PCI bus speed), and consumed 600mW at 3.3V. Testing of this design at 66MHz for use with next generation PCI buses is currently being performed.

A block diagram of the G2 chip is shown in Figure 1. This is a dataflow driven architecture with image data typically input as a serial raster scan. Control and data are entered byte-wise on common input pins. An integrated programmable control unit detects the control bytes and reconfigures the processor architecture by setting the variable length line buffers in the systolic array, multiplexing the image decimation/interpolation processors, initialization of the internal control sequencing, and possible execution of an automatic comparison and minimum detect across the 64 processing elements (PEs). Following reconfiguration, image data (8 bit grey-level pixels) is input until the image correlation operation is complete; no further instructions or external control are required. A database image (reference block) is correlated against a target image (search area). Nominally, an 8×8 PE array on a single chip can support up to a -3/+4 search area. Maximum image width is set by the length of the line buffers at 128 pixels at full resolution (no decimation). However since the variable length line buffers are individually programmable, a variety of other search ranges, search areas, and image sizes can be easily configured.

Each processing element implements a mean absolute difference (MAD) computation, consisting of an absolute difference (subtract, sign change/retain), and accumulation to the value previously stored in the PE. The subtractor is 8-bits wide; the accumulator is a 24-bit saturating adder which permits up to 23-bit differences (i.e. 128 × 256 images) to be handled at full accuracy. Much larger images can be handled if distinction of MAD values which exceed 23-bits is not needed (which is the case for most applications). The G2 chip also includes two processors for bilinear image interpolation (1 to 2 × 2) or decimation (2 × 2 to 1). In addition, a MAD minimum detection processor can be utilized to find both the address location and the minimum MAD value amongst the 64 PEs. Clocking and control signals are generated by an integrated programmable finite state machine.

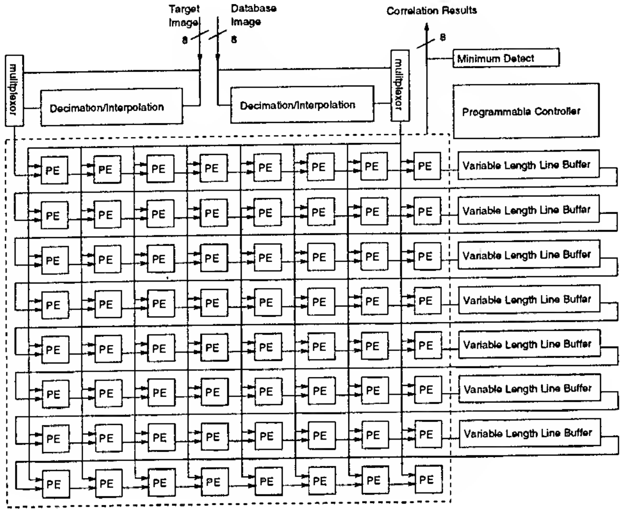


Figure 1: G2 Chip Architecture

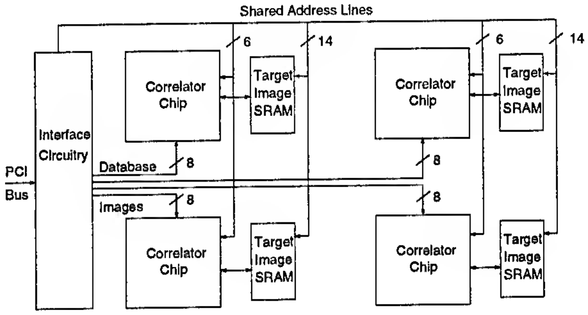


Figure 2: Coprocessor Board with 4 G2 Correlator Chips

In order to maximize the utilization of the PCI bus bandwidth, four G2 chips can be directly interfaced to the 32 bit PCI bus and to four local SRAM caches for the target images as shown in Figure 2. In the simplest configuration, all of the address lines to the four SRAM caches and four G2 chips are shared in common. Thus, the four G2 chips can directly used in parallel with common or distinct target images to be searched over a large database of search images. There are several other ways of programming the chip to enhance image search area, image size, etc. More details are given in [Bugeja and Yang, 1997]. In general a large number of these chips can be combined in parallel/cascade (cascade outputs are included at the end of the dataflow chain for this purpose) to realize very powerful configurations.

3 Applications and Benchmarks

Block matching is used in several different contexts; the G2 chip implements block matching of large templates over moderate search ranges (-3/+4 for a single chip operating in normal mode). This makes it ideally suited for applications involving image recognition. For example [Gilbert and Yang, 1993] describes a facial recognition system which uses the G1 chip as the central element in a system capable of recognizing a face in front of a camera by comparing against a large database of images; block matching is required for compensation of image misalignment and hence robust template registration. The G1 system acheived correlation speeds 4 times faster than a 150MHz Pentium running optimized assembly code, and could process 500 database images in 1 second over a search area of -2/+2; identification rates of 88% were measured using cross-validation. A four chip G2 system such as that shown in Fig. 2, on the other hand, implements a search area of -3/+4 for even better identification performance, and can process 8000 images in 1 second on a PCI board running at 33MHz, thus acheiving a 64× speedup over the 150MHz Pentium.

The G2 may also be used in other applications such as template search. The comparison of a large number of templates against a large search area is an essential task for many machine vision tasks but requires substantial processing time and computing resources. In a typical practical example, the templates might typically be fiducial marks of some sort, whilst the search area might typically be an image of a printed circuit board within which these fiducials are to be located for registration and detailed inspection. Unlike recognition applications, which are characterized by the correlation of database images against target images of only slightly larger size (e.g. 128×128 vs 135×135), search applications are typically characterized by large target templates compared to the images to be located within them.

We describe the application of the G2 for an example task involving the location of a 128×128 template within a 512×512 target. The scheme employs a two-level pyramid search involving search of a 32×32 template within a 128×128 version of the target (decimation by four) as a first step. The 128×128 template is broken down into 256 overlapping 39×39 blocks, and 256 correlations of the 32×32 template on each of these blocks are performed (taking time 256×45us on a single G2). Typically around 20 good matches are returned; 64×64 templates are then compared against a 256×256 target (decimation by 2), i.e. 20 64×64 correlations (20×151us). Finally, assuming around 4 good locations are returned, 4 correlations at full 128×128 resolution are performed to obtain the final best match (4×550us). A four chip G2 system can execute this algorithm in 4.2ms.

4 Conclusions

A coarse-grain reconfigurable coprocessor for compute and I/O intensive image processing tasks has been presented. The chip is designed around a core capable of carrying out block-matching type algorithms with high efficiency as a basic function; many other tasks can then be handled by appropriate programming of the core and other components available on chip. Typical applications, and the performance which systems constructed using the chip can acheive, have been briefly summarized.

References

[1] G. Knittel, L. Pocek, and J. Arnold, "A PCI-compatible FPGA-coprocessor for 2D/3D image processing," *Proceedings. IEEE Symposium on FPGAs for Custom Computing Machines,* pp. 136-145, in Napa Valley, CA, USA, 17-19 April. 1996.

[2] J. M. Gilbert and W. Yang, "A Real-Time Face Recognition System Using Custom VLSI Hardware," *Proceedings of the IEEE Workshop on Computer Architectures for Machine Perception,* pp. 58-66, in New Orleans, LA, USA, 15-17 Dec. 1993.

[3] H. Fujiwara, M. L. Liu, M. T. Sun, K. M. Yang, M. Maruyama, K. Shomura, and K. Ohyama, "An all-ASIC implementation of low bit-rate video codec," *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.2, No.2, pp. 123-134, June 1992.

[4] P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI Architectures for Video Compression - A Survey," *Proceedings of the IEEE,* Vol. 83, No. 2, pp. 220-246, Feb. 1995.

[5] A. Bugeja and W. Yang, "A Reconfigurable VLSI Coprocessing System for the Block Matching Algorithm," *IEEE Transactions on VLSI Systems,* to appear in Sep. 1997.

A Perspective 3D Formalism for Shape from Shading

Isaac Weiss
Center for Automation Research
University of Maryland, College Park, MD 20742-3275

Abstract

We develop a new general formalism for dealing with physics-based processes of image formation. Particular attention is given to the shape-from-shading problem, but other sensors, such as infrared, can also be treated. Unlike previous shape-from-shading methods, our formalism is three-dimensional and treats depth on an equal footing with the image coordinates. This makes it possible to treat the problem in perspective projection, rather than the usual orthographic projection. The formalism does not make specific assumptions about the surface, but makes it easy to incorporate a model-based assumption in the recovery of the shape. For example, we can assume that the shape is locally a quadric surface. This enables us to recover a surface without reliance on maximal-brightness points or on occluding boundaries, with their serious instability problems. The formalism is based on an adaptation of variational calculus, including Hamilton's equations and their invariance properties. In particular, scale-invariance is shown to be a useful property of the shading process.

1 Introduction

The shape-from-shading problem has usually been treated in a formalism introduced by Horn (e.g. [5]). This formalism assumed that we have a certain direction is space which is the "viewer direction", and shapes are orthographically projected on an image in this direction.

There are several problems with this formalism. First, the orthographic projection is only an approximation. The resulting error is more than purely geometric. It also distorts the grey level values at each point. The grey values depend on the angle between the surface normal and the direction of the emitted light. If this direction is distorted by assuming it is always the same direction, when in reality it is not, then the grey values will be distorted. Attempts at providing a non-orthographic treatment have been made (e.g. [6]) but they involved unnatural ("stereographic") projections and singularities, and had problems with integrability. Here we provide a full 3D perspective treatment of the shape-from-shading and similar problems. Our viewer is "isotropic": there is no particular viewer direction.

Another problem is that the usual formalism makes it hard to reconstruct the surface without knowledge of some special points or curves — for example, points of maximal brightness [4] or occluding contours [6]. However, such extremal points and curves are usually very hard to measure accurately. The image in a neighborhood around a maximal-brightness point is usually quite saturated, making the grey levels very unreliable. The grey levels near an occluding contour are also hard to measure accurately.

If we want to recover a surface from a patch of grey level values, without resort to extremal quantities, we need some additional knowledge about the surface. Several simple modeling assumptions have been tried. Some examples are the assumption of a locally spherical surface [7, 10], a locally cylindrical surface [19], or a surface with umbilical points everywhere [2]. However, these assumptions are either too restrictive, i.e. they make too strong an assumption about the surface, or not restrictive enough, thus not allowing us to find the surface uniquely. It is hard to find a modeling assumption which is "just right" in this sense, or even to determine what *is*

The author is grateful for the support of the Air Force Office of Scientific Research under Grant F49620-96-1-0355, and of the Office of Naval Research under Grant N00014-95-1-0521.

just right, namely how much information we need to add to the given shading information in order to recover the surface.

Our new formalism makes it easier to understand how much information is needed in addition to the shading and add it in a natural way. Because of the 3D structure of the formalism, we can add a modeling assumption that is most naturally expressed in a 3D representation. We do not have a preferred viewer direction or a preferred z axis; thus we can say things about the surface which are independent of such preferences. In one example that we use here, the surface is assumed to be locally quadric. This is equivalent to a smoothing assumption on the surface. With this assumption we are able to recover a non-Lambertian surface, in perspective projection, without the use of extremal points or curves. We do assume that the reflectance function is given, but there are no restrictions on this function. The method is non-iterative and there are no convergence problems.

Our new formalism is based on concepts from mathematical physics, mainly Hamilton's equation and its invariants. In [16, 17] we introduced the application of physics-like invariants to physical imaging processes. Here we make use of these concepts and extend the Hamiltonian formalism to 3D. The Hamiltonian formalism was also used for shading independently in [4] and [11], but without the use of invariance.

Invariance is important in vision in two aspects: the geometrical aspect and the physical one. While geometric invariants have been studied extensively lately (e.g. [9, 13, 14, 15, 18]), the physical ones still need to be fully exploited. Physics-like invariance is the analogy of the conservation of energy, momentum, etc. which are invariants of the laws of nature. The analogy can be very useful in vision.

2 Shading—The Physics Analogy

We use the shape-from-shading problem as the most difficult example that demonstrates the use of physics-like invariant methods. Other processes can be treated more simply.

In the shape-from-shading problem, we have data in the form of the image brightness $E(x,y)$, from which a surface $z(x,y)$ has to be recovered. This recovery depends on many unknowns, such as the surface function itself, the surface reflectance, the lighting, etc. As a way of simplifying the problem, the "reflectance map" was introduced by Horn, e.g. [5]. This is a function $R(p,q)$ that represents the amount of light emitted in the viewer (z) direction from a surface element with slope components p,q, in an orthographic projection. It contains only the two unknowns p,q which represent the slopes of the surface in the x and y directions. It is assumed that this function is known, i.e. that the light distribution and other factors are already built into R, and the only unknowns are the x and y slopes of the object at each point x,y. The light emitted from the surface element in the z direction falls on the image at point x,y, and gives rise to the image brightness $E(x,y)$ at that point. To a good approximation, this image brightness is proportional to the amount of light $R(p,q)$ that was emitted. Thus we can write the "image irradiance equation"

$$E(x,y) = R(p,q) \tag{1}$$

with the proportionality constant absorbed into E.

This deceptively simple looking equation is in fact a complicated partial differential equation, because p,q are the derivatives of the surface height $z(x,y)$ with respect to x,y: $p = \partial z/\partial x$, $q = \partial z/\partial y$. Our goal is to solve this equation for the surface function $z(x,y)$, given the brightness $E(x,y)$.

The irradiance equation can be solved by the method of *characteristics*, which is quite commonly used for first-order partial differential equations. We will briefly describe this method. In ordinary differential equations, one can start from a known initial point with given initial conditions, and propagate the solution from that point in infinitesimal steps, using the given equation. In partial differential equations, we need an initial curve rather than a point. From each point on this initial curve we grow a solution curve, the "characteristic" $x(t), y(t), z(t)$, going in steps of some parameter t (Figure 1). This characteristic is a solution of the equation along one curve. The collection of all characteristics, started from all points on the initial curve, constitutes the solution on the whole domain.

In most cases (including ours) we need the tangents to the characteristic curves as part of the solution, and thus the quantities $p(t), q(t)$ are added. The term "characteristic strips" rather than curves is often used for this situation.

In order to grow the characteristics for the irradiance equation (1), it has to be brought into a suitable form. This is done in [5]. We will later derive a generalized version. One obtains the four first-order coupled differential equations

$$\begin{aligned} \dot{x} &= R_p \qquad \dot{y} = R_q \\ \dot{p} &= E_x \qquad \dot{q} = E_y \end{aligned} \tag{2}$$

where the dot means differentiation with respect to the curve parameter t and the subscripts indicate differentiation.

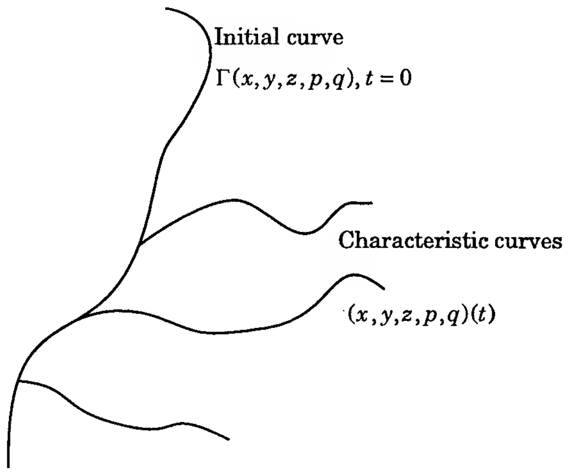


Figure 1: Method of characteristics

To grow a characteristic we proceed as follows. We start from a point x_0, y_0 on the given initial curve, with known slopes p_0, q_0. We substitute these values in the rhs of the above four equations, thus obtaining the values of the four derivatives $\dot{x}, \dot{y}, \dot{p}, \dot{q}$. We can then advance in a step of a small size Δt to a new point having coordinates and slopes $x_1 = x_0 + \dot{x}\Delta t$, etc. From this point we can iterate the process to form a whole characteristic curve. It remains to find $z(t)$, which can be done by integrating the equation

$$\dot{z} = pR_p + qR_q \tag{3}$$

in which the rhs is known.

The problem remains of finding the initial conditions. We can define an arbitrary initial curve $\Gamma(x, y)$, but we also need the values of p, q at each point on it. Thus we need some additional constraints, and there is no general way of obtaining them.

Common ways to deal with the problems are: i) Using points of maximal brightness (e.g. [4]). These are points where characteristics converge, so one needs initial conditions only at these points. However, the shading information around these points is very unstable, leading to unreliable characteristics [12]. ii) Using occluding boundaries (e.g. [6]). Here we know that the normal is tangent to the viewer's line of sight and that fact can be used as initial conditions. However, the shading information around an occluding boundary is also unstable, and the boundary can be occluded.

A way to overcome these problems is to use model-based knowledge about the surface in the process of recovering the surface. Our formalism makes it easy to do that because it treats the surface in a full 3D description rather than separating the depth information. For example, approximating the surface as locally quadric becomes quite natural using the ordinary 3D representation of a quadric. In addition, invariance properties such as scale or rotational invariance are a useful part of the formalism.

For our subsequent treatment we now recast the four equations (2) in a different form which is more easily handled by the calculus of variations. We define the Hamiltonian function H as

$$H = R - E$$

In this formulation, the irradiance equation (1) is $H = 0$, while the characteristic equations (2) can now be written in the form of Hamilton's equations

$$\dot{x} = \frac{\partial H}{\partial p} \qquad \dot{y} = \frac{\partial H}{\partial q} \tag{4}$$

$$\dot{p} = -\frac{\partial H}{\partial x} \qquad \dot{q} = -\frac{\partial H}{\partial y} \tag{5}$$

This form is similar to the one used to describe physical processes, and makes it possible to use the available knowledge in mathematical physics about invariants of such processes. It should be noted, however, that the similarity is only an analogy here, since our p, q are purely geometrical entities (slopes) while in physics they represent momentum. Our problem is in fact more general than the physics problem because there the kinetic energy is limited to the simple form $\mathbf{p}^2/2m$.

Continuing the physics analogy, the characteristics can be thought of as the trajectories of particles or fluid elements moving in time t with velocity $\dot{\mathbf{x}}$ and acceleration $\dot{\mathbf{p}}$. H represents the particle's "energy", comprised of kinetic energy R and potential energy $-E$. The equation $H = 0$ above means that the total energy is constant along the trajectory. Looking at Hamilton's equation above, we can see that an arbitrary constant can be added to the energy on each trajectory, since it will vanish in the differentiation in the rhs.

Beside finding invariants, another advantage of the Hamiltonian representation is that it enables us to deal with a more general irradiance equation than the one given in (1). This will enable us to deal with perspective projection and with non-shading physics-like processes. We have generalized the treatment in two ways [16, 17]:

(i) There is no need to separate the variables between two different functions $R(p, q)$, $E(x, y)$. Rather, we have one function $H(x, y, p, q)$ that depends on all four variables in a general way. In our shading example we will use a reflectance function R that depends on x, y as well as p, q.

Table 1: Relation between symmetries and invariants.

Symmetry under...	→	conservation law
Translation in time		energy
Translation in space		momentum
Rotation		angular momentum
Lorentz (space/time)		$E = mc^2$
Projective		–
Scale		scale momentum (?)

(ii) The variables p, q do not have to be slopes. They can be any variables for which the Hamilton equations make sense, and there can be many physics and physics-like processes that satisfy this equation. In the shading example they can be angles or 3D vectors. This is valuable because the reflectance laws are simpler (and more evidently symmetric) in terms of such variables.

3 Invariants and Symmetries

In this section we summarize the relation between symmetries of the physical laws and invariants, as it applies to shading (or shading-like) processes. Our full development is given in [16, 17].

In the classical example, energy and momentum are invariants of the equations of motion, namely they are constant as the particle moves along a trajectory. Energy conservation results from symmetry of the equations of motion with respect to translation in time, while momentum conservation results from symmetry of the motion equations with respect to translation in space. Similarly, conservation of angular momentum results from a rotational symmetry of the equations. By symmetry we mean that the *form of the equation* is invariant under the transformation. The *solution* is not generally symmetric and depends on the initial conditions. In other words, symmetry and invariance here are properties of the basic physical laws and not of any particular situation. Table 1 summarizes various symmetries used in physics and their corresponding invariants.

The main result concerning the relation between invariants and symmetries is Noether's theorem [8]. We give here only the result as it applies to the shading problem.

We deal with a general transformation of the coordinates and t, having r parameters w^s, $s = 1, \ldots, r$. For example, the rotation group in the plane has one parameter, the angle θ. This coordinate transformation can be written as

$$\bar{x}^i = x^i + \zeta^i_s dw^s, \qquad \bar{t} = t + \xi_s dw^s \qquad (6)$$

with ζ^i_s, ξ_s being coefficients characterizing the transformation. They generally depend on x^i, t. The summation convention is used. (In the 2D case x^i is x, y.)

If the reflectance function R is symmetric with respect to a transformation of the above type, Noether's theorem leads to the following "conservation law", as we showed in [16, 17]:

$$\frac{d}{dt}(p_i \zeta^i_s) = \frac{\partial E}{\partial w^s}$$

The rhs is a known quantity. In the physical analogy, the quantity $p_i \zeta^i_s$ in the lhs is the invariant such as energy or momentum, while the rhs is related to an external force. In the momentum case the above equation means that the change in momentum over time is proportional to the external force. Momentum is conserved in the absence of an external force.

The rhs can be written more explicitly in terms of the coordinates. For a general function of x^i we can write, using the chain rule,

$$\frac{\partial}{\partial w^s} = \frac{\partial}{\partial x^i} \zeta^i_s \qquad (7)$$

Thus the conservation law becomes

$$\frac{d}{dt}(p_i \zeta^i_s) = \frac{\partial E}{\partial x^i} \zeta^i_s \qquad (8)$$

An interesting property here is that we do not need to know the exact details of R to find the forms of various invariants, as R does not appear in the invariant equations above. The symmetry properties of R are sufficient.

It is clear that it is desirable to find transformations under which the reflectance function R, or the physical law that it describes, is symmetric. The coefficients ζ^i_s of this transformation will then be substituted into the conservation law (8) to find the invariant constraints. In [16, 17] we have done this for rotation and translation. Here we do it for the scaling transformation, which is the most important invariance in the shading case, because the reflectance function depends only on angles and not on scale. It has no physical analog because physical laws are not normally scale-invariant.

Scale invariance

The scaling transformation $\bar{x}^i = wx^i$ can be written, with a scaling parameter w, as

$$\bar{x}^i = x^i(1 + dw)$$

so that

$$\zeta^i = x^i$$

A scale change can be written, from (7), as

$$\frac{\partial}{\partial w} = \frac{\partial}{\partial x^i} x^i \tag{9}$$

Thus the conservation law (8) becomes

$$\frac{d}{dt}(p_i x^i) = \frac{\partial E}{\partial x^i} x^i \tag{10}$$

The rhs represents the scaling transformation of E, due to (9). We will later use a formulation in which E is scale-invariant and thus the rhs will vanish. Thus, the quantity $p_i x^i$ can perhaps be called the "scale momentum" or "magnification momentum". We will later expand on the use and geometrical significance of the above equations.

4 Non-Lambertian Surface in Perspective Projection

4.1 The isotropic viewer

Our formalism makes it possible to deal with shading in a perspective projection rather than the usual orthographic one. In fact, for a non-Lambertian surface, the treatment is easier and more natural in perspective projection.

In perspective projection, we can write the Hamiltonian $R - E$ in a way which is independent of the viewing direction, or the optical axis of the camera. This can be called an "isotropic" viewer. This is possible and desirable since the image brightness is basically a function of angles between the light rays and the surface normal, not of the optical axis. The physical construction of the camera does not need to be isotropic, i.e. the image sensor can remain flat. However $R - E$ can be represented as a function of angles which are independent of the optical axis. There is no "viewer's direction" here. This is unlike the usual orthographic projection which has a preferred direction.

Thus we can write the brightness, using the 3D vector $\mathbf{x} = (x, y, z)$, as

$$E(\frac{\mathbf{x}}{|\mathbf{x}|}) = E(\hat{\mathbf{x}})$$

i.e. the brightness at any point on the image depends on a 3D unit vector $\hat{\mathbf{x}}$ pointing from the origin (chosen as the camera optical center) towards the image. In this formulation, there is no need to distinguish between "world coordinates" and "camera coordinates" as in most other formulations. $\hat{\mathbf{x}}$ is the same on the image and on the object surface, for points connected by the same ray of light (Figure 2). The usual function $E(x, y)$ can easily be remapped into a function $E(\hat{\mathbf{x}})$. Here the choice of the z-axis direction is arbitrary and does not have to coincide with the camera axis (as long as it passes through the origin). Among other advantages, this can simplify the handling of singularities near the limb of the surface.

The surface will also be represented in a more isotropic way. Instead of $z(x, y)$ we describe the surface as an implicit function $f(x, y, z) = 0$. At each point of the surface we can define a vector perpendicular to it by

$$\mathbf{P} = \frac{df}{d\mathbf{x}} = (P, Q, U) \tag{11}$$

The perpendicularity can be seen by writing

$$df = \mathbf{P} \cdot d\mathbf{x} = 0. \tag{12}$$

When $\mathbf{x}, d\mathbf{x}$ are on the surface then $df = 0$ and $\mathbf{P}$, $d\mathbf{x}$ are perpendicular.

From this it is easy to calculate the surface gradient

$$\frac{\partial z}{\partial x} = -\frac{P}{U} = p, \qquad \frac{\partial z}{\partial y} = -\frac{Q}{U} = q \tag{13}$$

A unit normal to the surface can now be defined as

$$\hat{\mathbf{n}} = \frac{\mathbf{P}}{|\mathbf{P}|} = \frac{P, Q, U}{\sqrt{P^2 + Q^2 + U^2}} = \frac{-p, -q, 1}{\sqrt{p^2 + q^2 + 1}}$$

f, and thus $\mathbf{P}$, are determined here only up to a multiplicative factor g (not necessarily constant). This factor is eliminated from the gradient and normal expressions above. The characteristic equations will determine g up to the initial conditions.

The Hamilton equations (4),(5) with our generalized treatment (end of Section 2) are easily extended to the 3D isotropic case. We now write

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{P}}, \qquad \dot{\mathbf{P}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{14}$$

It is easily proved that equation (3) for $\dot{z}$, which was previously an extra equation, now comes out naturally from the 3D Hamilton equations. This is true for any irradiance function that depends on the normal $\hat{\mathbf{n}}$ rather than on $\mathbf{P}$. For any function $R(\hat{\mathbf{n}}, \cdots)$ we can write, from (14),

$$\dot{\mathbf{x}} = \frac{\partial}{\partial \mathbf{P}} R(\frac{\mathbf{P}}{|\mathbf{P}|}) = \frac{1}{|\mathbf{P}|}\frac{\partial R}{\partial \hat{\mathbf{n}}} - (\frac{\partial R}{\partial \hat{\mathbf{n}}} \cdot \mathbf{P})\frac{\mathbf{P}}{|\mathbf{P}|^3}$$

as can be verified componentwise. Multiplying by $\mathbf{P}$ we obtain

$$\mathbf{P} \cdot \dot{\mathbf{x}} = 0 \tag{15}$$

The geometrical meaning of this equation is that the direction of a characteristic in 3D is perpendicular to the surface normal. In other words, the characteristic stays on the surface. Dividing $\mathbf{P} \cdot \dot{\mathbf{x}} = 0$ by U

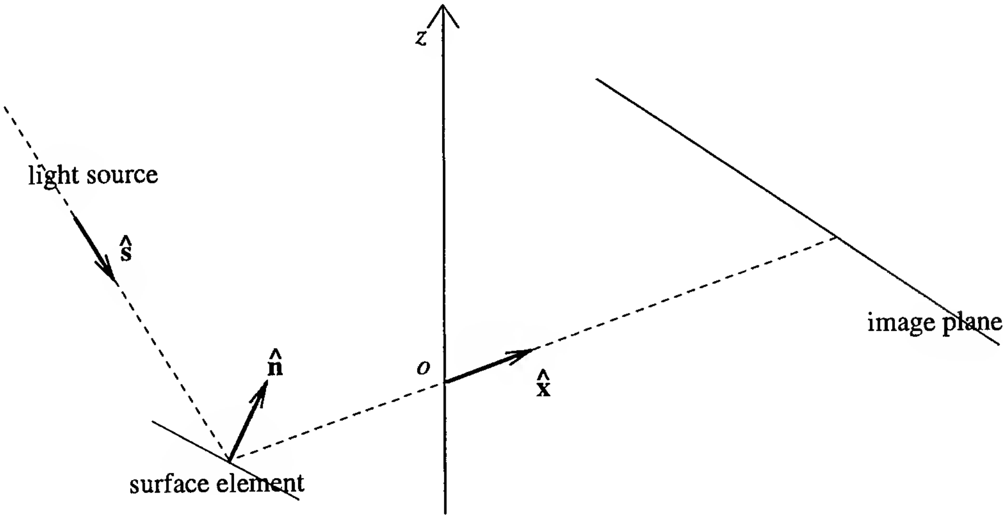


Figure 2: Isotropic viewer

we obtain (3), which was previously derived geometrically and now follows from Hamilton's equations.

The "physical" explanation is that because the "momentum" component perpendicular to the surface always equals 1 inside R, R cannot express changes of momentum $\mathbf{P}$ which are perpendicular to the surface; only changes tangent to the surface are meaningful. Since momentum change is proportional to force, the "force" driving the "particles" along characteristics is always tangent to the surface.

Eq. (15) has an important relation to the integrability of the system. From (12) we can see that (15) is equivalent to $df/dt = 0$. Thus (15) is a necessary condition for integrability to a surface, i.e. for obtaining a surface function $f(x,y,z) = 0$.

Another necessary condition is that an equation similar to (15) holds perpendicular to the characteristic, while (15) holds along it. This can be written, with a parameter s going across a characteristic, as

$$\frac{df}{ds} = \mathbf{P}\cdot\frac{d\mathbf{x}}{ds} = 0 \tag{16}$$

This equation is independent of the Hamilton equations. We will use it to close a system of equations for moving across characteristics.

4.2 Generalized reflectance function

With the above formalism it is easy to treat a general surface under perspective projection. For simplicity, we carry out the treatment for the case in which the light source is rotationally symmetric around the direction $\hat{s}$. However, this restriction is immaterial to the development and we will later show how to remove it. Another simplification to be removed later is the assumption of constant albedo.

The Hamiltonian can be written as

$$H = R(\hat{\mathbf{n}}\cdot\hat{s}, \hat{\mathbf{n}}\cdot\hat{\mathbf{x}}) - E(\hat{\mathbf{x}}) = 0 \tag{17}$$

The term $\hat{\mathbf{n}}\cdot\hat{\mathbf{x}}$ in R represents the light emitted from the surface. It means that the emitted light intensity is a function of the angle between the surface normal $\hat{\mathbf{n}}$ and the direction $\hat{\mathbf{x}}$ of a light ray in perspective projection (Figure 2). This is in accordance with the general law of reflectance.

Since R now depends on $\mathbf{x}$, we can no longer use the simple shading equations (2); the more general Hamilton equations (14) have to be used. This is an advantage of the Hamiltonian formalism and the generalized treatment at the end of Section 2.

The Hamilton equations can be developed, using the identity

$$\frac{\partial\hat{\mathbf{n}}}{\partial\mathbf{P}} = \left(\frac{I}{|\mathbf{P}|} - \frac{\mathbf{P}*\mathbf{P}}{|\mathbf{P}|^3}\right) = \frac{1}{|\mathbf{P}|}(I - \hat{\mathbf{n}}*\hat{\mathbf{n}})$$

where I is the unit matrix and $*$ denotes the "outer product" of vectors. Similarly for $\hat{\mathbf{x}}$. We obtain

$$\begin{aligned}\dot{\mathbf{x}} = &\frac{\partial R}{\partial\mathbf{P}} = \frac{\partial R}{\partial(\hat{s}\cdot\hat{\mathbf{n}})}\frac{1}{|\mathbf{P}|}(\hat{s} - (\hat{s}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}) \\ &+ \frac{\partial R}{\partial(\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})}\frac{1}{|\mathbf{P}|}(\hat{\mathbf{x}} - (\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}})\end{aligned} \tag{18}$$

$$\begin{aligned}\dot{\mathbf{P}} = &-\frac{\partial(R-E)}{\partial\mathbf{x}} = -\frac{\partial R}{\partial(\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})}\frac{1}{|\mathbf{x}|}(\hat{\mathbf{n}} - (\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})\hat{\mathbf{x}}) \\ &+ \frac{1}{|\mathbf{x}|}\left(\frac{\partial E}{\partial\hat{\mathbf{x}}} - \left(\frac{\partial E}{\partial\hat{\mathbf{x}}}\cdot\hat{\mathbf{x}}\right)\hat{\mathbf{x}}\right)\end{aligned} \tag{19}$$

When the reflectance depends on the surface albedo, than R has an explicit dependence on $\hat{\mathbf{x}}$:

$$R = R(\hat{s}\cdot\hat{\mathbf{n}}, \hat{\mathbf{x}}\cdot\hat{\mathbf{n}}, \hat{\mathbf{x}})$$

In this case the first Hamilton equation above is unchanged. In the second equation, E is replaced by $E - R$.

4.3 Invariance properties

The fundamental invariance property of the above equations is scale-invariance. This holds because both the R and E parts of the Hamiltonian depend only on angles and not on distances. This is unlike most physical laws.

The brightness E is now scale-invariant, so by (9) we can write

$$\frac{\partial E}{\partial \mathbf{x}} \cdot \mathbf{x} = \frac{\partial E}{\partial w} = 0$$

with w being a scaling parameter. Thus the scale-invariance equation (10) simplifies to

$$\frac{\partial}{\partial t}(\mathbf{x} \cdot \mathbf{P}) = 0$$

I.e., the "scale momentum" $\mathbf{x} \cdot \mathbf{P}$ is conserved along characteristics. This can also be proved directly from Hamilton's equations. The scale-invariance of this momentum can be seen from the definition of $\mathbf{P}$ as a gradient $df/d\mathbf{x}$ (13). When the coordinates are scaled by w, the momentum is scaled by $1/w$, keeping the product $\mathbf{x} \cdot \mathbf{P}$ invariant.

In most cases it can be assumed that the characteristics in some region of the image meet at some point, such as a point of maximal brightness. (For our purposes we do not need to know the location of that point.) Even if two characteristics do not meet, they can still be assigned the same scale momentum, because in this case we are free to adjust $\mathbf{P}$ at the initial points by an arbitrary multiplicative factor. Thus all characteristics start with the same amount of scale momentum and conserve the same momentum. Therefore the scale momentum is conserved *across* characteristics as well as along them:

$$\frac{\partial}{\partial s}(\mathbf{x} \cdot \mathbf{P}) = 0 \tag{20}$$

Subtracting (16) from (20) we obtain

$$\frac{\partial \mathbf{P}}{\partial s} \cdot \mathbf{x} = 0 \tag{21}$$

The analogous equation for the t-direction can be proved in the same way, and also directly by multiplying (19) scalarly by $\mathbf{x}$.

All the above equations are independent of any rotational or translational symmetries that the system may have, and thus the system does not have to possess these symmetries in order for us to move across characteristics.

4.4 Moving across a characteristic

Here we develop the equations necessary to move across a characteristic in our 3D formalism. We deal with a general surface. In the next section we will close the system of equations for a locally quadric surface.

We assume that the values of $\mathbf{x}, \mathbf{P}$ are known at some initial point. We denote by t, s the parameters along and across a characteristic, respectively. These parameters are defined on the *3D surface*, not in the 2D image projection. We will denote derivatives wrt the parameters by subscripts, e.g. $\mathbf{x}_t, \mathbf{x}_s$. (The t-derivatives are the same as the dot derivatives above).

From the condition that $\mathbf{x}_s, \mathbf{x}_t$ are tangent to the surface (15),(16) we have

$$\mathbf{x}_t \cdot \mathbf{P} = 0 \tag{22}$$

$$\mathbf{x}_s \cdot \mathbf{P} = 0 \tag{23}$$

These two equations are equivalent to $f_s = f_t = 0$, and ensure the integrability of the system. The last one is independent of the Hamilton equations and will enable us to close the system of equations. Next, we define the arc length along s by

$$\mathbf{x}_s \cdot \mathbf{x}_s = \mathbf{x}_t \cdot \mathbf{x}_t \tag{24}$$

This definition is quite arbitrary, but it is reasonable to choose $\mathbf{x}_s$ so it will scale the same way as $\mathbf{x}_t$. This equation holds only along the initial curve which is parametrized by s. On subsequent s-curves we will not be free to define the arclength; the Euclidean distance between characteristics changes as we go along them even though their distance in the s parameter remains constant.

Another useful dot product is derived by multiplying (18) by $\mathbf{x}_t$ and using (22):

$$\mathbf{x}_t \cdot \mathbf{x}_t = \mathbf{F} \cdot \mathbf{x}_t \tag{25}$$

with

$$\mathbf{F} = \frac{\partial R}{\partial(\hat{\mathrm{s}} \cdot \hat{\mathbf{n}})} \frac{1}{|\mathbf{P}|} \hat{\mathrm{s}} + \frac{\partial R}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} \frac{1}{|\mathbf{P}|} \hat{\mathbf{x}}$$

Another relation can be easily obtained from (18):

$$\hat{\mathrm{s}} \cdot \mathbf{x}_t \times \mathbf{P} = \frac{\partial R}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} \hat{\mathrm{s}} \cdot \hat{\mathbf{x}} \times \hat{\mathbf{n}} \equiv G \tag{26}$$

The rhs vanishes for a Lambertian or for a circular surface.

To find more equations we take derivatives of the above equations. Differentiating (23) wrt t we obtain

$$\mathbf{x}_{st} \cdot \mathbf{P} + \mathbf{x}_s \cdot \mathbf{P}_t = 0 \tag{27}$$

Next we differentiate (22), (25), (26) wrt s to obtain

$$\mathbf{x}_{st} \cdot \mathbf{P} + \mathbf{x}_t \cdot \mathbf{P}_s = 0 \tag{28}$$

$$2\mathbf{x}_t \cdot \mathbf{x}_{st} = \mathbf{F} \cdot \mathbf{x}_{st} + \mathbf{F}_s \cdot \mathbf{x}_t \tag{29}$$

$$\hat{\mathrm{s}} \cdot \mathbf{x}_{st} \times \mathbf{P} + \hat{\mathrm{s}} \cdot \mathbf{x}_t \times \mathbf{P}_s = G_s \tag{30}$$

with

$$\begin{aligned}\mathbf{F}_s =& \frac{\partial^2 R}{\partial(\hat{\mathrm{s}} \cdot \hat{\mathbf{n}})^2} \frac{\hat{\mathrm{s}} \cdot \hat{\mathbf{n}}_s}{|\mathbf{P}|} \hat{\mathrm{s}} + \frac{\partial^2 R}{(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})^2} \frac{\hat{\mathbf{x}}_s \cdot \hat{\mathbf{n}} + \hat{\mathbf{x}} \cdot \hat{\mathbf{n}}_s}{|\mathbf{P}|} \hat{\mathbf{x}} \\ &+ \frac{\partial R}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} \frac{1}{|\mathbf{P}|} \hat{\mathbf{x}}_s - \frac{\mathbf{P} \cdot \mathbf{P}_s}{|\mathbf{P}|^2} \mathbf{F}\end{aligned}$$

$$\begin{aligned}G_s =& \frac{\partial^2 R}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})^2} \hat{\mathrm{s}} \cdot (\hat{\mathbf{x}} \times \hat{\mathbf{n}})(\hat{\mathbf{x}}_s \cdot \hat{\mathbf{n}} + \hat{\mathbf{x}} \cdot \hat{\mathbf{n}}_s) \\ &+ \frac{\partial R}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} \hat{\mathrm{s}} \cdot (\hat{\mathbf{x}}_s \times \hat{\mathbf{n}} + \hat{\mathbf{x}} \times \hat{\mathbf{n}}_s)\end{aligned}$$

and

$$\hat{\mathbf{n}}_s = \frac{\mathbf{P}_s}{|\mathbf{P}|} - \frac{(\mathbf{P} \cdot \mathbf{P}_s)\mathbf{P}}{|\mathbf{P}|^3}, \qquad \hat{\mathbf{x}}_s = \frac{\mathbf{x}_s}{|\mathbf{x}|} - \frac{(\mathbf{x} \cdot \mathbf{x}_s)\mathbf{x}}{|\mathbf{x}|^3} \tag{31}$$

Finally we rewrite (21):

$$\mathbf{P}_s \cdot \mathbf{x} = 0 \tag{32}$$

The above seven equations (23),(24),(27)-(30), (32) form a system of equations for the nine components of $\mathbf{x}_s, \mathbf{x}_{st}, \mathbf{P}_s$. Eqs. (28),(29),(30) are equivalent to the s-derivative of (18), but they are simpler. The last five equations have the form of a scalar product, which is explicitly rotationally invariant.

To close the system of equations we need to make some assumption about the surface. This is done is the next section.

We will mention for completeness the relation between the tangency conditions (22),(23) (related to the scale invariance) and the irradiance equation (17):

$$(\mathbf{x}_t \cdot \mathbf{P})_s - (\mathbf{x}_s \cdot \mathbf{P})_t = \mathbf{x}_t \cdot \mathbf{P}_s - \mathbf{x}_s \cdot \mathbf{P}_t = \frac{\partial H}{\partial s} = 0$$

This equation is independent of Hamilton's equations (which leave us with an integration constant which depends on s) but it does follow from the irradiance equation.

4.5 A locally quadric surface

Here we assume that the surface is locally quadric. This means that third-order derivatives are assumed to be relatively small at any point.

Quadric polynomials were used in [3]. There it was the brightness that was approximated by quadrics rather than the surface itself (as in our case), and a segmentation of the surface was needed.

A quadric surface can be described using a 3×3 symmetric matrix A as

$$f = g[(\mathbf{x} - \mathbf{x}_0)\mathsf{A}(\mathbf{x} - \mathbf{x}_0) + C] = 0$$

where $\mathbf{x}_0$ is the center, C is a constant, and $g(\mathbf{x}) \neq 0$ is some arbitrary function. By the definition of the momentum we have

$$\mathbf{P} = \frac{\partial f}{\partial \mathbf{x}} = 2g\mathsf{A}(\mathbf{x} - \mathbf{x}_0)$$

Multiplying by $\mathbf{x}$, we see that the scale momentum $\mathbf{x} \cdot \mathbf{P}$ is a function of g. The Hamilton equation will set g so that the scale momentum is constant. For a central quadric, namely $\mathbf{x}_0 = 0$, it is easily shown that constant scale momentum yields a constant g.

A central conic is sufficient for our purposes. This is because, when observed from any direction z, a general quadric surface $z(x, y)$ is indistinguishable from a central quadric up to second derivatives. I.e., the derivatives $z_x, z_y, z_{xx}, z_{xy}, z_{yy}$ can be set to any given values, for any $\mathbf{x}_0$. (This is to be distinguished from the derivatives of f, of which there are nine up to second order.) Thus having the center at the origin (the optical center) is not a significant restriction. Of course, the center does not need to be "inside" a convex object. For example, an object can be approximated locally as a hyperboloid, whose center is "outside" the body and lies at the origin.

We can thus simply set $g = 1/2$ and obtain

$$\mathbf{P} = \mathsf{A}\mathbf{x}$$

We are interested in finding $\mathbf{x}_s, \mathbf{P}_s$. Differentiating wrt s we obtain three simple relations between the vector components:

$$\mathbf{P}_s = \mathsf{A}\mathbf{x}_s \tag{33}$$

However, two of these relations follow from the general equations in the previous subsection. Multiplying by $\mathbf{x}$ we get

$$\mathbf{x} \cdot \mathbf{P}_s = \mathbf{x}\mathsf{A}\mathbf{x}_s = \mathbf{P} \cdot \mathbf{x}_s$$

and similarly

$$\mathbf{x}_t \cdot \mathbf{P}_s = \mathbf{P}_t \cdot \mathbf{x}_s$$

Both of these equations are general to any surface and follow from the previous derivations. A third equation is obtained by multiplying (33) by $\mathbf{x}_{tt}$:

$$\mathbf{x}_{tt} \cdot \mathbf{P}_s = \mathbf{x}_{tt}\mathsf{A}\mathbf{x}_s = \mathbf{P}_{tt} \cdot \mathbf{x}_s \tag{34}$$

This does not follow from previous equations and it characterizes the central quadric. We can see that this relation contains second-order derivatives of E appearing in $\mathbf{P}_{tt}$, unlike all previous equations.

Another equation can be obtained by differentiating (33) wrt t and multiplying by $\mathbf{x}_{tt}$:

$$\mathbf{x}_{tt} \cdot \mathbf{P}_{st} = \mathbf{P}_{tt} \cdot \mathbf{x}_{st} \tag{35}$$

Here $\mathbf{P}_{st}$ can be written as a function of $\mathbf{x}_s, \mathbf{P}_s$ by differentiating Hamilton's equation (19) wrt s. Thus we have two more equations (34),(35) to close the system we had before.

In summary, we have obtained nine linear equations (23),(24),(27)–(30), (32),(34),(35) for the nine unknowns $\mathbf{x}_s, \mathbf{P}_s, \mathbf{x}_{st}$. It should be noted that these nine equations have to be solved at every point of the initial curve only, not at every point of the surface.

It remains to find the "curvatures" $\mathbf{x}_{ss}$. Differentiating (23),(24) wrt s we get

$$\mathbf{x}_{ss} \cdot \mathbf{P} + \mathbf{x}_s \cdot \mathbf{P}_s = 0$$

$$\mathbf{x}_{ss} \cdot \mathbf{x}_s = \mathbf{x}_t \cdot \mathbf{x}_{st}$$

It is also easy to prove from the general equations of the previous section that

$$\mathbf{x}_s \cdot \mathbf{P}_s = \mathbf{x} \cdot \mathbf{P}_{ss}$$

$$\mathbf{x}_s \cdot \mathbf{p}_s = \mathbf{x}_{ss} \cdot \mathbf{P}$$

$$\mathbf{x}_t \cdot \mathbf{P}_{ss} + \mathbf{x}_{st} \cdot \mathbf{P}_s = \mathbf{x}_s \cdot \mathbf{P}_{st} + \mathbf{x}_{ss} \cdot \mathbf{P}_t$$

These five equations are independent of any particular assumption such as the quadric assumption. Since we have six unknowns $\mathbf{x}_{ss}, \mathbf{P}_{ss}$, we need an additional equation. Eq. (33) can be differentiated with respect to s and multiplied by $\mathbf{x}_{tt}$, which yields one relation independent of the above:

$$\mathbf{x}_{tt} \cdot \mathbf{P}_{ss} = \mathbf{P}_{tt} \cdot \mathbf{x}_{ss}$$

We thus have six linear equations for $\mathbf{x}_{ss}, \mathbf{P}_{ss}$. Given these quantities, we can proceed along the curve $\mathbf{x}(s)$ across the characteristic by steps of second-order accuracy. The accumulation of steps produces first-order accuracy of the curve at its far end from the starting point. Again, we need to do this only on the initial curve, not on the whole surface.

4.6 General light sources and albedo

In the previous section we assumed that the light source was rotationally symmetric around $\hat{\mathbf{s}}$. This was done only for simplicity. We now describe the modifications needed to remove this restriction.

For a general light source our generalized reflectance function can be written as

$$R = \int r(\hat{\mathbf{n}} \cdot \hat{\mathbf{s}}, \hat{\mathbf{n}} \cdot \hat{\mathbf{x}}, \hat{\mathbf{x}}) d\hat{\mathbf{s}}$$

with the integration being carried over all the light directions $\hat{\mathbf{s}}$, and with r representing the intensity of the light as a function of the direction $\hat{\mathbf{s}}$. The argument $\hat{\mathbf{x}}$ above (separately from $\hat{\mathbf{n}} \cdot \hat{\mathbf{x}}$) represents the dependence of the reflectance on the surface albedo.

Thus, in all the equations above that involve R, R will be replaced by r and the terms of the equations will be integrated over $\hat{\mathbf{s}}$. In the case of variable albedo, Hamilton's equation (19) is also modified, as mentioned earlier, by replacing E with $E - R$ to account for the explicit dependence of R on $\mathbf{x}$. Scale-invariance is preserved and all the previous equations are valid.

Special attention is needed to handle the equations with terms that vanish due to rotational symmetry, namely (26) and its s-derivative, (30). Multiplying Hamilton's equation (18) vectorially by $\mathbf{P}$ and integrating we obtain

$$\dot{\mathbf{x}} \times \mathbf{P} = \left(\int \frac{\partial r}{\partial(\hat{\mathbf{s}} \cdot \hat{\mathbf{n}})} \hat{\mathbf{s}} d\hat{\mathbf{s}} \right) \times \hat{\mathbf{n}} + \hat{\mathbf{x}} \times \hat{\mathbf{n}} \int \frac{\partial r}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} d\hat{\mathbf{s}}$$

In analogy to the symmetric case, we multiply this equation by a weighted average of the light direction, $\hat{\mathbf{s}}^*$. This is defined to be parallel to the vector integral above:

$$\hat{\mathbf{s}}^* \times \int \frac{\partial r}{\partial(\hat{\mathbf{s}} \cdot \hat{\mathbf{n}})} \hat{\mathbf{s}} d\hat{\mathbf{s}} = 0 \tag{36}$$

This determines the two free components of the unit vector $\hat{\mathbf{s}}^*$. We obtain from the previous equation

$$\hat{\mathbf{s}}^* \cdot \dot{\mathbf{x}} \times \mathbf{P} = G \equiv \hat{\mathbf{s}}^* \cdot \hat{\mathbf{x}} \times \hat{\mathbf{n}} \int \frac{\partial r}{\partial(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})} d\hat{\mathbf{s}}$$

This replaces (26), and its s-derivative replaces (30):

$$\hat{\mathbf{s}}^* \cdot \mathbf{x}_{st} \times \mathbf{P} + \hat{\mathbf{s}}^* \cdot \mathbf{x}_t \times \mathbf{P}_s + \hat{\mathbf{s}}^*_s \cdot \mathbf{x}_t \times \mathbf{P} = G_s$$

with $\hat{\mathbf{s}}^*_s$ being calculated from the s-derivative of (36).

5 Conclusions

We have presented a general method of dealing with physical and physics–like imaging processes. The formalism is based on Hamilton's equations, and has several advantages:

i) Our formalism is fully 3D, unlike the usual treatment that separates the depth from the rest of the variables. This makes it possible to use perspective projection in cases that were previously treated only by less general projections, e.g. orthographic in shading, or paraperspective in texture. To do this we have introduced the "isotropic viewer" representation of a camera and incorporated it into our generalized "reflectance function".

ii) Our formalism makes it possible to find invariants of the imaging processes. We show how such invariants can be used in the recovery of the surface. The invariants result from symmetries of the imaging processes. The shading process is the hardest one in which to find symmetry, because it generally has only scale invariance. Other processes such as infrared imaging have rotational or other symmetries. Texture can also be treated like a shading problem without a light source [1]. All these cases can be handled by our formalism as special cases.

iii) The 3D formalism makes it easy to incorporate model-based geometric knowledge about the surface. We showed as an example the approximation of a surface locally as a quadric. This can replace the reliance on singular points and boundaries with their serious instability problems.

Implementation is feasible because we need only second-order derivatives of the brightness. Experiments are under way.

References

[1] J. Aloimonos and M. Swain, "Shape from Pattern: Regularization", *Int. J. Computer Vision*, **2**, 171–187, 1988.

[2] A. Blake, A. Zisserman and G. Knowles, "Surface Descriptions from Stereo and Shading", *Image and Vision Computing*, **3**, 183–191, 1985.

[3] R.M. Bolle and D.B. Cooper, "Bayesian Recognition of Local 3D Shape by Approximating Image Intensity Functions with Quadric Polynomials", *IEEE T-PAMI*, **6**, 418–429, 1984.

[4] P. Dupuis and J. Oliensis, "An Optimal Control Formulation and Related Numerical Methods for a Problem in Shape Reconstruction", *Annals of Applied Probability*, **4**, 287–346, 1994.

[5] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.

[6] K. Ikeuchi and B.K.P. Horn, "Numerical Shape from Shading and Occluding Boundary", *Artificial Intelligence*, **17**, 141–184, 1981.

[7] C.-H. Lee and A. Rosenfeld, "Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System", *Artificial Intelligence*, **26**, 125–143, 1985.

[8] D. Lovelock and H. Rund, *Tensors, Differential Forms, and Variational Principles*, Dover, New York, 1975.

[9] J.L. Mundy and A. Zisserman, Eds: *Geometric Invariance in Machine Vision*, MIT Press, Cambridge, MA, 1992.

[10] A.P. Pentland, "Local Shading Analysis", *IEEE T-PAMI*, **6**, 170–187, 1984.

[11] E. Rouy and A. Tourin, "A Viscosity Solution Approach to Shape-from-Shading", *SIAM J. Numer. Anal.*, **29**, 867–884, 1992.

[12] B.V.H. Saxberg, "An Application of Dynamical Systems Theory to Shape from Shading", *Proc. DARPA Image Understanding Workshop*, 1089–1104, 1989,

[13] I. Weiss, "Projective Invariants of Shapes", *Proc. DARPA Image Understanding Workshop*, 1125–1134, 1988.

[14] I. Weiss, "Noise-Resistant Invariants of Curves", *IEEE T-PAMI*, **15**, 943–948, 1992.

[15] I. Weiss, "Geometric Invariants and Object Recognition", *Int. J. Computer Vision*, **10**, 201–231, 1993.

[16] I. Weiss, "Invariants for Recovering Shape from Shading", in *Application of Invariance in Computer Vision*, Lecture Notes in Computer Science No. 825, Springer-Verlag, Berlin, 1993.

[17] I. Weiss, "Physics-like Invariants for Vision", *Proc. Workshop on Physics-Based Modeling in Computer Vision*, 55-61, 1995; *Proc. DARPA Image Understanding Workshop*, 1413–1421, 1994.

[18] I. Weiss, "Local Projective and Affine Invariants", *Annals of Mathematics and Artificial Intelligence*, **13**, 203-225, 1995.

[19] R.L. Wildey, "Radarclinometry", *Earth, Moon and Planets*, **36**, 217–247.

Image Segmentation and Labeling Using the Polya Urn Model

A. Banerjee† P. Burlina† F. Alajaji‡

†Center for Automation Research
University of Maryland
College Park, MD 20742-3275

‡Department of Mathematics and Engineering
Queen's University
Kingston, ON K7L 3N6, Canada

Abstract

We propose a segmentation method based on Polya's model for contagious phenomena. An initial segmentation is obtained using a Maximum Likelihood (ML) estimate or the Nearest Mean Classifier (NMC). The resulting clusters are then subjected to a morphological process operating like the development of an infection to yield segmentation of the image into homogeneous regions. This process is implemented using contagion urn processes and generalizes Polya's scheme by allowing spatial interactions. The urn mixture model provides a fuzzy representation of the pixel label. The composition of the urns is iteratively updated by assuming a Markovian relationship between neighboring pixel labels. The asymptotic behavior of this process is examined. Examples of the application of this scheme to the segmentation of Synthetic Aperture Radar (SAR) images and Magnetic Resonance Images (MRI) are provided.

1 Introduction

We describe a segmentation method using contagion urn schemes that rely on a modified version of the Polya-Eggenberger sampling process [8]. This biologically inspired sampling procedure was originally designed to model the development of contagious phenomena.

Many approaches to segmentation have been studied. Unsupervised segmentation approaches include Nearest Mean Classification (NMC) and the branch-and-bound procedure [3]. Supervised methods generally proceed by formulating statistical model assumptions for the image formation and region generation processes. Maximum likelihood (ML) or maximum *a posteriori* (MAP) estimation is then used for segmentation. Examples of such approaches abound in the literature [4; 6; 11]. Techniques modeling images as Markov random fields (MRFs) have been extensively investigated [4]. MRFs attempt to represent spatial dependencies and the MRF-Gibbs equivalence allows for the computation of the maximum *a posteriori* (MAP) estimate of the original image.

This paper models images using urn processes. The motivation for employing urn schemes is twofold: First, urn processes can generate Markov chains as well as MRFs [5]. Second, urn schemes are of particular interest because they provide a natural representation for fuzzy image labeling. Therefore, they constitute an attractive generative process for the underlying image regions which exhibit strong spatial dependencies. Our work is related to the Gibbs sampling procedure [4], preserving key features of the Gibbs sampler but using instead a contagion sampling scheme. The spatial dependencies of the pixel labels are captured by the contagious behavior which promotes smoothing of the image into contiguous regions. The urn process for segmentation is related to relaxation labeling algorithms [10], except that the urn process is not deterministic.

We begin by applying either an ML or NMC segmentation technique to the image. The contagion process is then applied to the image labels. In this scheme, each pixel is represented by an urn with a mixture of balls of different colors, one color for each class label. A neighborhood system is also defined on each pixel. The balls of the urns of the neighborhood system are then combined to determine the next states of the urns. The iterative nature of the algorithm incorporates temporal memory, while the inclusion of the neighboring urns in the update promotes spatial contagion. Moreover, the neighborhood system is modified, pending the existence of an edge element in the neighborhood. This is done to preserve edges by confining the propagation of similar class labels within closed boundaries.

This paper is organized as follows: The initial NMC and ML segmentations are presented in Section 2. The contagion-based smoothing process is then described in Section 3. In Section 4, the stochastic properties of the resulting image process are discussed. Finally, experimental results on SAR and MR images are shown in Section 5.

2 Initial Segmentation

When no a priori information on the image statistics is available, general clustering algorithms such as NMC are usually applied. In the NMC method, an initial arbitrary labeling is used from which centroids of the feature vectors of each class are computed. Next, all samples are reclassified to the cluster corresponding to the nearest mean, and the centroids are recomputed. This process is iterated until a stopping criterion is met [3].

In contrast, when a stochastic model for the image can be justified, it is possible to apply ML segmentation. The conditional distribution of the image, i.e., the form

$$p(X_s/C_s = l, C_r; r \in N_s^k) \tag{1}$$

is assumed. Here, C_s is the label for pixel s, C_r represents the pixel labels of N_s^k, the k^{th} order neighborhood of pixel s, and X_s is the given image data [4].

For segmentation purposes, we estimate the pixel labels by assuming that the conditional probability of each class label, i.e. $p(X_s/C_s = l)$, is governed by a multivariate Gaussian distribution on the second-order neighborhood N_s^2.

After obtaining the parameters of the different classes, the ML test determines the label for each pixel in the image. The ML decision rule is

$$\hat{l} = \arg\max_l \ p(X_s/C_s = l, C_r; r \in N_s^2). \tag{2}$$

The above schemes do not capture the statistics and connectedness of local regions. Since the ML test assumes that each pixel label is equally likely throughout the image, it produces a noisy segmentation. This assumption is incorrect, for in a local region dominated by one class, the dominant class has a higher prior probability than the other classes. Such contextual information is not taken into account in either the ML or NMC estimate of the pixel labels.

This drawback is usually addressed within the framework of MAP segmentation. The MAP estimate of the class label $\hat{l}$ for a pixel given the observed image X_s is

$$\hat{l} = \arg\max_l \ p(C_s = l/C_r; r \in N_s^2, X_s) \tag{3}$$

Indeed, it can be shown that maximizing $p(C_s = l/C_r; r \in N_s^2, X_s)$ is equivalent to maximizing $p(C_s = l/C_r; r \in N_s^2)p(X_s/C_s = l)$.

If segmentation of the image into homogeneous regions is desired, it is intuitively appealing to model the prior distribution $p(C_s = l/C_r; r \in N_s^2)$ using an MRF, as the MRF model relates the label of a pixel to the labels of its neighboring pixels [6].

If the prior is modeled as an MRF, the Gibbs-MRF equivalence can be exploited by techniques such as simulated annealing (SA) or other stochastic relaxation methods to derive the MAP estimate [6].

Unfortunately, techniques such as simulated annealing have high computational costs. Indeed, convergence to the MAP estimate is possible only when impractically slow annealing schedules are followed. Instead, we propose to replace the annealing step by an urn contagion process to model the spatial dependencies between neighboring pixels.

3 Image Sampling with Contagion

The labeled image is described by an urn process. Each pixel in the image is represented by an urn containing a mixture of balls of different colors representing the classes. The proportion of each class in the urn indicates the similarity of the pixel to the class. The urn representation is therefore a fuzzy representation of the segmented image. At each iteration the current urn is modified by re-sampling with contagion, a process that is inspired by the original urn sampling process introduced by Polya and Eggenberger in [8].

3.1 Temporal Contagion

The work reported in [8] introduced the following urn scheme as a model for the spread of a contagious disease through a population. An urn originally contains T balls, of which W are white and B are black ($T = W + B$). Successive draws from the urn are made; after each draw, $1 + \Delta$ ($\Delta > 0$) balls of the same color as was just drawn are returned to the urn. Let $\rho = W/T$ and $\delta = \Delta/T$. Define the binary process $\{Z_n\}_{n=0}^{\infty}$ as follows:

$$Z_n = \begin{cases} 0, & \text{if the } n^{th} \text{ ball drawn is white;} \\ 1, & \text{if the } n^{th} \text{ ball drawn is black.} \end{cases}$$

It can be shown that the process $\{Z_n\}$ is stationary and non-ergodic [7; 9]. The urn scheme has infinite memory, in the sense that each previously drawn ball has an equal effect on the outcome of the current draw.

3.2 Temporal and Spatial Contagion

The urn sampling scheme proposed in this paper incorporates both temporal and spatial contagion. Instead of representing an image by a finite lattice of pixels, we consider an image as a finite lattice of urns. In the "one-dimensional" urn sampling described above, the effect of each sample propagates through time. For the "two-dimensional" case, the sampled ball at each iteration must depend not only on the composition of the pixel's urn, but also on the compositions of the neighboring urns to encourage contagious behavior. Thus, we need to allow for spatial interactions at each time instant by *associating* the urns of the neighboring pixels in the determination of the newly sampled ball.

3.3 A Fuzzy Image Labeling Representation

The following presentation considers, without loss of generality, a binary labeling problem. Let $I_n = [p_n^{(i,j)}]$ be a binary label image of size $K \times L$, where $p_n^{(i,j)} \in \{0,1\}$ is the label of pixel (i,j) at iteration n, $n = 0, 1, \ldots,$ $(i,j) \in \mathcal{I}$ where

$$\mathcal{I} : \{(i,j) : i = 0, \ldots, K-1; j = 0, \ldots, L-1\}.$$

To each pixel we associate an urn $u_n^{(i,j)} : (B_n^{(i,j)}, W_n^{(i,j)})$ with each pixel (i,j) at time n, where $B_n^{(i,j)}$ and $W_n^{(i,j)}$ are respectively the number of black and white balls in the urn. With this representation we define a membership function for each pixel as

$$m_F^B(p_n^{(i,j)}) = \frac{B_n^{(i,j)}}{(B_n^{(i,j)} + W_n^{(i,j)})}.$$

3.4 An Algorithm for Segmentation with Spatial Contagion

The general class of algorithms for the contagion-based smoothing process can be described as follows:

- **Initialization**

Let I_0 be an initial segmentation (at time index $n = 0$). For each pixel (i,j), the initial urn composition $u_0^{(i,j)} = (B_0^{(i,j)}, W_0^{(i,j)})$ is obtained by computing the relative frequencies of white and black pixels in a spatial neighborhood centered on (i,j). For this work, the second order (3×3) neighborhood system for each pixel is adopted.

- **Iterative Image Sampling**

For $n > 0$, the urn composition of each pixel (i,j) is updated by sampling from a combination of the participating urns $\mathcal{V}_{n-1}^{(i,j)}$ with $\mathcal{V}_{n-1}^{(i,j)} : \{u_{n-1}^{(r,s)} : (r,s) \in N_q^k\}$, where N_q^k is the neighborhood system defined as in [4]:

$$N_q^k : \{q = (r,s) \in \mathcal{I} : (i-r)^2 + (j-s)^2 \leq k\}.$$

A simple, yet effective, sampling procedure is as follows: the urn $u_n^{(i,j)}$ for pixel $p_n^{(i,j)}$ is updated by first combining the balls of $u_{n-1}^{(i,j)}$ and the N neighboring urns:

$$\mathcal{C}_{n-1}^{(i,j)} = \text{ASSOCIATE}(\mathcal{V}_{n-1}^{(i,j)}). \qquad (4)$$

The ASSOCIATE function forms a collection of balls, $\mathcal{C}_{n-1}^{(i,j)}$, from the urns of the neighborhood. Examples of the ASSOCIATE function include grouping the urns of $\mathcal{V}_{n-1}^{(i,j)}$ into a "super" urn or sampling one ball from each urn to form the collection. Furthermore, the neighborhood may be modified if an edge element exists in that neighborhood; if so, those neighboring urns which lie on the other side of the edge are excluded. This is necessary to preserve edges and limit contagion to local areas.

Next, an operation on the new collection of balls, $\mathcal{C}_{n-1}^{(i,j)}$, is performed, i.e.

$$Z_n^{(i,j)} = \text{SELECT}(\mathcal{C}_{n-1}^{(i,j)}). \qquad (5)$$

The SELECT function may determine the next state of the urns by sampling one ball from $\mathcal{C}_{n-1}^{(i,j)}$ or by taking the majority class of $\mathcal{C}_{n-1}^{(i,j)}$.

We denote by $Z_n^{(i,j)}$ the outcome of the SELECT function:

$$Z_n^{(i,j)} = \begin{cases} 0, & \text{if the } n^{th} \text{ ball drawn is white;} \\ 1, & \text{if the } n^{th} \text{ ball drawn is black.} \end{cases}$$

If $Z_n^{(i,j)} = 0$, add Δ white balls to urn $u_n^{(i,j)}$; if $Z_n^{(i,j)} = 1$, add Δ black balls to urn $u_n^{(i,j)}$.

This yields a new urn composition for each pixel, given by

$$u_n^{(i,j)} : \begin{cases} W_n^{(i,j)} = W_{n-1}^{(i,j)} + (1 - Z_n^{(i,j)}) * \Delta, \\ B_n^{(i,j)} = B_{n-1}^{(i,j)} + (Z_n^{(i,j)}) * \Delta. \end{cases}$$

The above procedure is iterated until $n = N$. At time N, the final composition of each individual urn $u_N^{(i,j)}$, $(i,j) \in \mathcal{I}$ determines the final labeling of the image. As described above, each urn represents a fuzzy membership function on the pixel labels.

4 Statistical Properties

4.1 Temporal Contagion

The resulting sequence of generated images exhibits both spatial and temporal dependencies represented by a Markovian relationship in terms of the urns $u_n^{(r,s)}$; more specifically:

$$Pr\{u_n^{(i,j)} | U_{n-1}, U_{n-2}, \ldots, U_0\} = Pr\{u_n^{(i,j)} | \mathcal{V}_{n-1}^{(i,j)}\},$$

where $U_n : [u_n^{(i,j)}]$ is the urn matrix associated with I_n, and $\mathcal{V}_{n-1}^{(i,j)}$ is the set of participating urns defined in the previous section.

Consider the original Polya sampling scheme. The asymptotic properties of the joint distribution can be characterized in the "one-dimensional" case, i.e., when all spatial interactions are inhibited at each sampling step. In this case, it can be shown [9] that the proportion of white balls in each urn after the n^{th} trial $\rho_n^{(i,j)}$, where

$$\rho_n^{(i,j)} = \frac{\rho + \left(Z_1^{(i,j)} + Z_2^{(i,j)} + \cdots + Z_n^{(i,j)}\right)\delta}{1 + n\delta},$$

is a martingale [2] and admits a limit Z as the number of draws increases indefinitely. Indeed, $\rho_n^{(i,j)}$ (or equivalently the sample average $\frac{1}{n}\sum_{k=1}^{n} Z_k^{(i,j)}$) converges with probability 1 to Z [2]. This limiting proportion Z is a continuous random variable with support the interval $(0,1)$ and beta probability density function with parameters $(\rho/\delta, (1-\rho)/\delta)$:

$$f_Z(z) = \begin{cases} \frac{\Gamma(1/\delta)}{\Gamma(\rho/\delta)\Gamma((1-\rho)/\delta)} z^{\frac{\rho}{\delta}-1}(1-z)^{\frac{1-\rho}{\delta}-1}, \\ if\ 0 < z < 1; \\ 0, \text{otherwise}. \end{cases}$$

$\Gamma(\cdot)$ is the gamma function described by

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \quad \text{for } x > 0.$$

The behavior of this pdf can be interpreted as follows: Assuming $\delta = 1$ for simplicity, if the original fraction of white balls in the urn is close to 1, then the limiting distribution of $W_n^{(i,j)}$ will be skewed towards 1. A similar behavior is obtained for the case when ρ is close to 0. Therefore, the limiting pattern will reflect the underlying probability

$$Pr\left(p_1^{(i,j)} = x\right) = \rho^x(1-\rho)^{(1-x)}.$$

For the M-ary labeling case, the above observations generalize with convergence to the Dirichlet distribution [5].

4.2 Temporal and Spatial Contagion

We examine the asymptotic behavior of two examples of a general urn sampling scheme for segmentation.

- **Method 1**

Consider sampling from the "super" urn. Restating the problem, suppose there are N urns in the neighborhood of pixel X_s, each *initially* with b_i black balls and w_i white balls, and $b_i + w_i = T$ for all i, $i = 1, 2, \ldots, N$. We put the contents of all N urns into a "super" urn, sample one ball, and add Δ balls of the same color into the urn of pixel X_s. The following properties are easily derived:

The probability of sampling exactly k black balls from n iterations of the "super" urn is

$$Pr(X = k) = \binom{n}{k} \frac{B(\alpha + k, \beta + n - k)}{B(\alpha, \beta)} \tag{6}$$

where $\alpha = \sum_i \frac{b_i}{\Delta}$, $\beta = \sum_i \frac{w_i}{\Delta}$, and the beta function $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

The above process can be regarded as being generated by a sequence of independent Bernoulli trials with parameter Z, where Z is random with beta distribution. In fact, it is identical with different parameters to the Polya-Eggenberger distribution in the "one-dimensional" case given above.

The average number of black balls in the "super" urn at any given time is expressed as

$$E[B_n] = \sum_j b_j \frac{NT + n\Delta}{NT}. \tag{7}$$

Therefore, the average proportion of black balls in the "super" urn is

$$E\left[\frac{B_n}{(NT + n\Delta)}\right] = \sum_j \frac{b_j}{NT} \tag{8}$$

Remarkably, the average proportion of black balls in the "super" urn at any time instant equals the original proportion of black balls. The above results show that the composition of the urn is highly dependent on the original proportion of the balls. Eventually, the majority class of the urns in a given neighborhood will spread and dominate the population of balls in that neighborhood. Therefore, we conclude that this urn sampling scheme will reinforce the majority class in a local spatial neighborhood; it constitutes a positive-feedback system that yields limiting patterns of the self-reinforcing type [1]. The contagion effectively models the Markovian dependencies of the pixel labels.

- **Method 2**

This second example is described as follows: We sample one ball from each of the urns in pixel $X_{(i,j)}$'s neighborhood, $\mathcal{V}_{n-1}^{(i,j)}$. From this collection of balls, we compute the majority class, denoted by $Z_n^{(i,j)}$. We update urn $u_n^{(i,j)}$ in the same manner described in the previous section, i.e.

$$u_n^{(i,j)} : \begin{cases} W_n^{(i,j)} = W_{n-1}^{(i,j)} + (1 - Z_n^{(i,j)}) * \Delta, \\ B_n^{(i,j)} = B_{n-1}^{(i,j)} + (Z_n^{(i,j)}) * \Delta. \end{cases}.$$

The composition of urn $u_n^{(i,j)}$ is governed by the Polya-Eggenberger distribution as explained above. Eventually, the initial majority class of each urn in the neighborhood will dominate its composition.

It is difficult to find a general closed-form expression for $P(Z_n^{(i,j)} = k)$, the probability that class k is the majority of the individual samples. The difficulty arises because we are trying to find the majority of a set of samples of a non-i.i.d. process. Hence, we resort to heuristic arguments. Experimental results will be given in the next section.

5 Experimental Results

For segmentation of SAR imagery, we start with ML segmentation. As shown in Figure 1(a), the resulting labeling is spotty, a characteristic of the ML segmentation technique. Application of simulated annealing generates a contiguous segmentation of the image (Figure 1(b)). Likewise, Figure 1(c) shows that ten iterations of the urn sampling scheme operating on the ML segmentation yield an image labeled into locally homogeneous regions. The SAR image used in this example is from Lincoln Laboratory's ADTS SAR data, which is fully polarimetric with 1 foot resolution.

Whereas simulated annealing achieves segmentation by optimizing a function (the MAP pixel label estimate), modified urn schemes smooth the segmented image by morphologically processing the pixel labels. Since the Polya urn schemes model contagious behavior in a population, modified urn schemes allow dominant pixel labels to propagate within local regions, analogous to diffusion methods for segmentation. The advantage of using the urn scheme lies in the reduction of the computational complexity of the segmentation algorithm. We avoid the time and computational costs of simulated annealing by employing a simpler algorithm.

To segment the MR images, we obtain an initial segmentation by NMC. The inherent noise of this image modality leads to a speckled segmentation. The contagion urn process then operates on the pixel labels to produce a smoother segmentation. The outputs after one and ten iterations are shown in Figures 2(b) and 2(c), respectively. Note that the edges are preserved by limiting contagion to local areas. An edge map, computed by the Canny edge detector, is employed to modify each pixel's sampling neighborhood to prohibit sampling over different region types.

In both cases, sampling method 2 is implemented; one ball is sampled from each of the urns of the neighborhood and a majority rule is applied to determine the next state of the urns. Each urn is initialized with ten balls, and Δ, the number of balls added at each iteration, is 2.

6 Conclusion

In this paper, we have illustrated how modified Polya urn sampling schemes can be implemented for image segmentation. Given an initial speckled segmentation, the contagion process obtains a smoother segmentation into homogeneous regions by its Markovian properties. Two general properties incorporate temporal and spatial contagion. First, iterative updating is required for temporal contagion. Second, sampling from neighboring urns, similar to the Gibbs sampler, is necessary to encourage spatial contagion.

Further lines of research include the evaluation of optimal values for the parameter δ, the ratio of Δ to the initial number of balls in an urn. If δ is too high, the segmentation is over-smoothed; if it is too low, the algorithm may not converge to the appropriate segmentation. As mentioned above, the initial composition of the urns determines to a great extent the outcome of the contagion process. Therefore, finding an appropriate method of initializing the urn composition is critical to accurately segmenting the image.

References

[1] W. B. Arthur, Y. M. Ermoliev, and Y. M. Kaniovski, "Path-Dependent Processes and the Emergence of Macro-Structures," *European Journal of Operational Research*, pp. 294–303, 1987.

[2] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 2, Wiley, 1971.

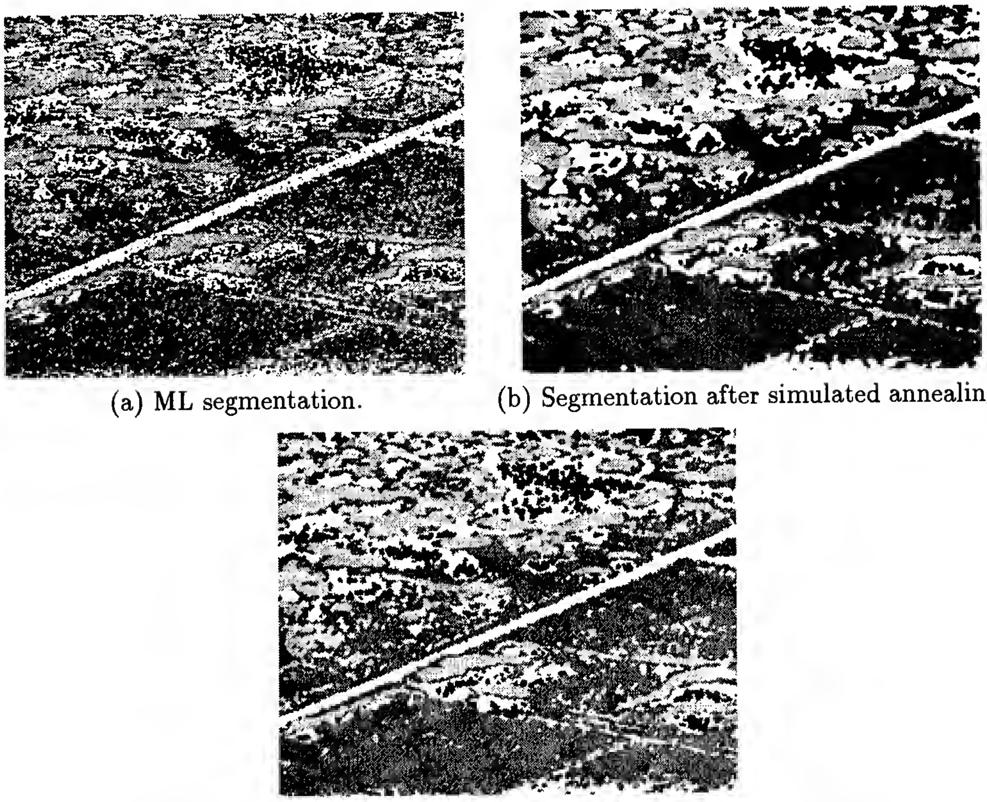
[3] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, 1990.

[4] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 721–741, 1984.

[5] N. Johnson and S. Kotz, *Urn Models and Their Application*, Wiley, 1977.

[6] P. A. Kelly, H. Derin, and K. D. Hartt, "Adaptive Segmentation of Speckled Images Using a Hierarchical Random Field Model," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1628-1641, 1988.

[7] G. Polya, "Sur Quelques Points de la Théorie des Probabilités," *Ann. Inst. H. Poincaré*, Vol. 1, pp. 117–161, 1931.



(a) ML segmentation.

(b) Segmentation after simulated annealing.

(c) After urn process.

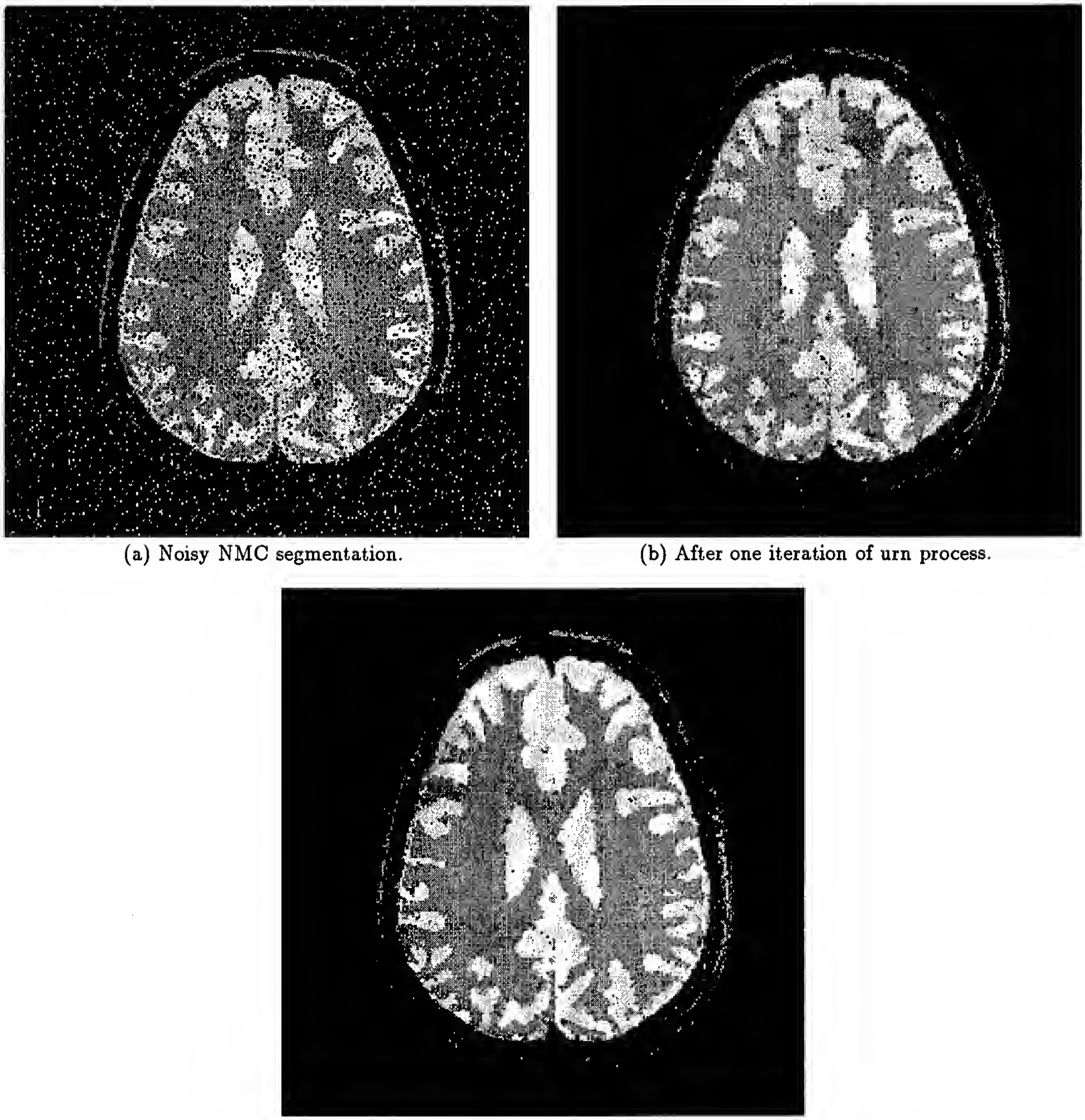
Figure 1: Segmentation of ADTS SAR image after ten iterations of SA and urn process

[8] G. Polya and F. Eggenberger, "Uber die Statistik Verketteter Vorgänge," *Z. Angew. Math. Mech.*, pp. 279–289, 1923.

[9] G. Polya and F. Eggenberger, "Sur l'Interpretation de Certaines Courbes de Fréquences," *Comptes Rendus C. R.*, pp. 870–872, 1928.

[10] S. Ranade and A. Rosenfeld, "Point Pattern Matching by Relaxation," *Pattern Recognition*, Vol. 12, pp. 269–275, 1980.

[11] E. Rignot and R. Chellappa, "Segmentation of Polarimetric Synthetic Apperture Radar Data," *IEEE Trans. on Image Processing*, Vol. 1, pp. 281–300, 1992.



(a) Noisy NMC segmentation.

(b) After one iteration of urn process.

(c) After ten iterations of urn process.

Figure 2: Segmentation of MR Images using urn process with inhibition

Analysis of Reconstruction from Multiple Views

Cornelia Fermüller and Yiannis Aloimonos
Computer Vision Laboratory
Center for Automation Research
Institute for Advanced Computer Studies
and Computer Science Department
University of Maryland
College Park, MD 20742

Abstract

If 3D rigid motion is estimated with some error a distorted version of the scene structure will in turn be computed. Of computational interest are those regions in space where the distortions are such that the depths become negative, because in order to be visible the scene has to lie in front of the image. The stability analysis for the structure-from-motion problem presented in this paper investigates the optimal relationship between the errors in the estimated translational and rotational parameters of a rigid motion, that results in the estimation of a minimum number of negative depth values. The input used is the value of the flow along some direction, which is more general than optic flow or correspondence. For a planar retina it is shown that the optimal configuration is achieved when the projections of the translational and rotational errors on the image plane are perpendicular. Furthermore, the projections of the actual and the estimated translation lie on a line passing through the image center. For a spherical retina, given a rotational error, the optimal translation is the correct one, while given a translational error the optimal rotational error is normal to the translational one at an equal distance from the real and estimated translations. The proofs, besides illuminating the confounding of translation and rotation in structure from motion, have an important application to ecological optics, explaining differences between planar and spherical eye or camera designs in motion and shape estimation.

1 Introduction

The general problem of structure from motion is defined as follows: given a number of views of a scene, to recover the rigid transformations between the views and the structure (shape) of the scene. In the field of computational vision a lot of effort has been devoted to this problem because it lies at the heart of several applications in pose estimation, recognition, calibration, and navigation [Faugeras, 1992; Hartley, 1994; Maybank, 1993].

While many solutions have been proposed, they become problematic in the case of realistic scenes and most of them degrade ungracefully as the quality of the input deteriorates. This has motivated research on the stability of the problem; Daniilidis and Spetsakis [1996] contains an excellent survey of existing error analyses. In summary, it can be concluded that the majority of the existing analyses attempt to model the errors in either the 3D motion estimates or the depth estimates, and due to the large number of unknowns in the problem, they deal with restricted conditions such as planarity of the scene or non-biasedness of the estimators. Notably absent in published efforts is an account of the systematic nature of the errors in the depth estimates due to errors in the 3D motion estimates.

In this paper an approach that is independent of any algorithm or estimator is taken. Due to the geometry of image formation any spatiotemporal representation in the image is due to the 3D motion and the structure of the scene. If the 3D motion can be estimated correctly, the structure can be derived correctly using the equations of image formation. However, an error in the estimation of the 3D motion will result in the computation of a distorted version of the actual scene structure. Of computational interest are those regions in space where the distortions are such that the depths become negative. Not considering any scene interpretation, the only fact we know about the scene is that for it to be visible it has to lie in front of the image and thus the depth estimates have to be positive. Therefore the number of image points whose corresponding scene points would yield negative values due to erroneous 3D motion estimation should be kept as small as possible. This is the computational principle behind the error analysis presented in this paper. In particular, the following questions are studied. Assuming there is an error in the estimation of the rotational motion components, what is the error in the translational components that leads to a minimization of the number of negative depth values computed? Similarly, if there is an error in the translational motion estimates, which rotational error will result in the smallest number of negative depth values? The analysis is carried out for a complete field of view as perceived by an imaging sphere, and for a restricted field of view on a constrained image plane.

2 Overview and problem statement

2.1 Prerequisites

We consider an observer moving rigidly with translation $\mathbf{t} = (U, V, W)$ and rotation $\boldsymbol{\omega} = (\alpha, \beta, \gamma)$ in a stationary environment, and an image formation based on perspective projection. If the image is formed on a sphere of radius f (i.e., $\mathbf{r}\cdot\mathbf{r} = f^2$), the 2D velocity $\dot{\mathbf{r}}$ at an image point $\mathbf{r} = (x, y, z)$ corresponding to a scene point $\mathbf{R} = (X, Y, Z)$ is

$$\dot{\mathbf{r}} = \frac{\mathbf{v}_{\text{tr}}(\mathbf{r})}{|\mathbf{R}|} + \mathbf{v}_{\text{rot}}(\mathbf{r}) = -\frac{1}{|\mathbf{R}|f}(\mathbf{r}\times(\mathbf{r}\times\mathbf{t})) - \boldsymbol{\omega}\times\mathbf{r} \tag{1}$$

where $|\mathbf{R}|$, being the norm of $\mathbf{R}$, denotes the range, and $\frac{\mathbf{v}_{\text{tr}}(\mathbf{r})}{|\mathbf{R}|}$ and $\mathbf{v}_{\text{rot}}(\mathbf{r})$ denote the translational and rotational components respectively.

Similarly, if the image is formed on a plane orthogonal to the Z axis at distance f from the nodal point, the 2D image velocity is

$$\begin{aligned}\dot{\mathbf{r}} &= \frac{\mathbf{v}_{\text{tr}}(\mathbf{r})}{Z} + \mathbf{v}_{\text{rot}}(\mathbf{r}) \\ &= -\frac{1}{Z}(\mathbf{z}_0\times(\mathbf{t}\times\mathbf{r})) \\ &\quad + \frac{1}{f}(\mathbf{z}_0\times(\mathbf{r}\times(\boldsymbol{\omega}\times\mathbf{r})))\end{aligned} \tag{2}$$

where $\mathbf{z}_0$ denotes a unit vector in the direction of the Z axis.

The component of the flow u_n along any direction $\mathbf{n}$ is therefore

$$u_n = \dot{\mathbf{r}}\cdot\mathbf{n} = \frac{\mathbf{v}_{\text{tr}}}{Z}\cdot\mathbf{n} + \mathbf{v}_{\text{rot}}\cdot\mathbf{n} \quad \text{or} \quad \frac{\mathbf{v}_{\text{tr}}}{|\mathbf{R}|}\cdot\mathbf{n} + \mathbf{v}_{\text{rot}}\cdot\mathbf{n} \tag{3}$$

Since only the direction of translation $\mathbf{t}/|\mathbf{t}|$ and the depth (range) of the scene up to a scaling factor can be obtained, that is, $\frac{Z}{|\mathbf{t}|}\left(\frac{|\mathbf{R}|}{|\mathbf{t}|}\right)$, we set $|\mathbf{t}| = 1$.

2.2 Distorted space

Let us assume there is an error in the estimation of the five motion parameters. As a consequence there will also be errors in the estimation of depth (range) and thus a distorted version of the space will be computed. A convenient way to describe the distortion of space is to sketch it through surfaces in space which are distorted by the same multiplicative factor, the iso-distortion surfaces.

In the following, in order to distinguish between the various estimates, we use letters with hat signs to represent the estimated quantities $(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, |\hat{\mathbf{R}}|, \hat{Z}, \hat{\mathbf{v}}_{\text{tr}}, \hat{\mathbf{v}}_{\text{rot}})$ and unmarked letters to represent the actual quantities $(\mathbf{t}, \boldsymbol{\omega}, |\mathbf{R}|, Z, \mathbf{v}_{\text{tr}}, \mathbf{v}_{\text{rot}})$. The subscript "$\epsilon$" is used to denote errors, where we define $\boldsymbol{\omega} - \hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_\epsilon$ and $\mathbf{v}_{\text{rot}} - \hat{\mathbf{v}}_{\text{rot}} = \mathbf{v}_{\text{rot}_\epsilon}$.

The estimated depth or range can be obtained from (3) as

$$\hat{Z}\ (\text{or } |\hat{\mathbf{R}}|) = \frac{\hat{\mathbf{v}}_{\text{tr}}\cdot\mathbf{n}}{\dot{\mathbf{r}}\cdot\mathbf{n} - \hat{\mathbf{v}}_{\text{rot}}\cdot\mathbf{n}}$$

and thus on the image sphere we get

$$|\hat{\mathbf{R}}| = |\mathbf{R}| \cdot \left(\frac{(\mathbf{r}\times(\mathbf{r}\times\hat{\mathbf{t}}))\cdot\mathbf{n}}{(\mathbf{r}\times(\mathbf{r}\times\mathbf{t}))\cdot\mathbf{n} + f|\mathbf{R}|(\boldsymbol{\omega}_\epsilon\times\mathbf{r})\cdot\mathbf{n}}\right) \tag{4}$$

From (4) it can be seen that $\left|\hat{\mathbf{R}}\right|$ can be expressed as a multiple of $|\mathbf{R}|$, where the multiplicative factor, which we denote by D, the distortion factor, is given by the term inside the brackets. Thus the distortion factor is

$$D = \frac{(\mathbf{r}\times(\mathbf{r}\times\hat{\mathbf{t}}))\cdot\mathbf{n}}{(\mathbf{r}\times(\mathbf{r}\times\mathbf{t}))\cdot\mathbf{n} + f|\mathbf{R}|(\boldsymbol{\omega}_\epsilon\times\mathbf{r})\cdot\mathbf{n}} \tag{5}$$

Similarly, on the image plane we can interpret the estimated depth as a multiple of the actual depth with distortion D, where

$$D = \frac{-f(\mathbf{z}_0\times(\hat{\mathbf{t}}\times\mathbf{r}))\cdot\mathbf{n}}{-f(\mathbf{z}_0\times(\mathbf{t}\times\mathbf{r}))\cdot\mathbf{n} + Z(\mathbf{z}_0\times(\mathbf{r}\times(\boldsymbol{\omega}_\epsilon\times\mathbf{r})))\cdot\mathbf{n}} \tag{6}$$

Equations (5) and (6) describe, for any fixed direction $\mathbf{n}$ and any distortion factor D, a surface in space. Any such surface is to be understood as the locus of points in space which are distorted in depth (range) by the same factor D, if the corresponding image measurements are in direction $\mathbf{n}$.

Figure 1 illustrates a family of iso-distortion surfaces corresponding to the same gradient direction but different distortion factors D. As can be seen the iso-distortion surfaces of a family intersect in a curve, and they change continuously as we vary D. Thus all the space between the 0 distortion surface and the

$-\infty$ distortion surface (which is also the $+\infty$ distortion surface) is distorted by a negative distortion factor.

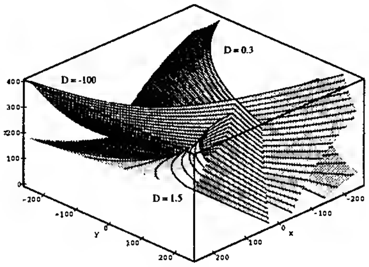


Figure 1: Family of iso-distortion surfaces.

2.3 Description of results

In the forthcoming sections we employ a geometric statistical model to represent the negative depth values. We assume that the scene in view lies within a certain depth (range) interval. The flow representation vectors in the image are in different directions, and we assume some distribution for their directions. Our focus is on the points in space which for a 3D motion estimate yield negative depth (range) estimates.

For every direction $\mathbf{n}$ the points in space with negative depth estimates cover the space between the 0 and $-\infty$ distortion surfaces within the range covered by the scene. Thus for every direction we obtain a 3D subspace, covering a certain volume. The sum of all volumes for all directions, normalized by the flow distributions considered, represents a measure of the likelihood that negative depth values occur. We call it the "negative depth volume" or "negative range volume." The idea behind our error analysis lies in the minimization of this negative depth (range) volume—that is, we are interested in the relationship between the translational and rotational motion errors that minimizes this volume.

In our analysis we assume that the flow directions are uniformly distributed in every direction and at every depth (range) between a minimum value $Z_{\min}(|\mathbf{R}_{\min}|)$ and a maximum value $Z_{\max}(|\mathbf{R}_{\max}|)$.

In summary, as an answer to the question about the coupling of motion errors, the following results are obtained:

A: If we take the whole sphere as the imaging surface and we assume an error in the estimation of rotation, then the direction of translation that minimizes the negative depth volume is the correct direction of translation.

The practical implication of this result is that 3D motion estimation is most easily accomplished for a complete field of view, as provided by an imaging sphere. A working system (biological or artificial) is usually equipped with an inertial sensor which provides rotational information, though probably with some error. On the basis of this information, the best one can do to estimate the remaining translation is to assume that the flow field obtained by subtracting the estimated rotation is purely translational and apply a simple algorithm designed for only translation [Horn and Weldon, Jr., 1988].

Estimation of purely translational motion is much simpler than estimation of complete 3D rigid motion, which requires techniques that decouple the translation from the rotation in some way. Thus insects with spherical eyes, such as bees and flies, have a big advantage in the task of 3D motion estimation.

B: On the other hand, if we assume a certain error in the estimation of translation on a spherical image, we find that the vector of the rotational error $\boldsymbol{\omega}_\epsilon$ lies on the same geodesic as the real translation $\mathbf{t}$ and the estimated translation $\hat{\mathbf{t}}$ at equal distance from both, that is, $(\mathbf{t} + \hat{\mathbf{t}}) \times \boldsymbol{\omega}_\epsilon = 0$ ($\mathbf{t}$ and $\hat{\mathbf{t}}$ are unit vectors).

C: Considering as imaging surface a plane of limited extent, we find that the translational and rotational errors are perpendicular to each other. Using the notation $\frac{Uf}{W} - \frac{\hat{U}f}{\hat{W}} = x_{0_\epsilon}$ and $\frac{Vf}{W} - \frac{\hat{V}f}{\hat{W}} = y_{0_\epsilon}$, this means that $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon}$. If we fix the rotational error $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$, this provides us with a constraint on the direction of the translational error.

D: If we fix the translational error $(x_{0_\epsilon}, y_{0_\epsilon})$ we obtain the same constraint, and in addition we find that $\gamma_\epsilon = 0$. Furthermore, if we only fix the amount of translational error we find that $\frac{U}{V} = \frac{\hat{U}}{\hat{V}}$.

These results are in accordance with experimental observations and with proofs derived under particular simplifying assumptions.

The importance of the results obtained for the plane also lies in their consequences for shape estimation. They can be translated into the statement that planar retinas with high resolution at the center are advantageous in the computation of shape. As will be shown in Section 5, if $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon}$ and $\frac{U}{V} = \frac{\hat{U}}{\hat{V}}$, near the fixation center for any depth Z, the distortion factor is approximately the same for every flow direction! This means that all scene points of the same depth are distorted by the same factor and thus a depth map is derived whose level contours are the correct ones!

3 Analysis on the sphere

3.1 Fixed rotational error

As a parameterization for expressing the orientations $\mathbf{n}$ we choose the following:

As shown in Figure 2a, let $\boldsymbol{\omega}_\epsilon$ be parallel to the x axis and let $\mathbf{s}$ be the set of all the unit vectors in the yz plane with $\mathrm{s} = (0, \sin\chi, \cos\chi)$ and χ in the interval $[0 \ldots \pi]$. The flow directions $\mathbf{n}$ at every point are defined as $\mathbf{n} = \frac{\mathbf{s}\times\mathbf{r}}{|\mathbf{s}\times\mathbf{r}|}$. In this parameterization, $\mathbf{n}$ takes on every possible orientation in the tangent plane at every point, but not all orientations are treated equally. In order to obtain a uniform distribution we must perform some normalization.

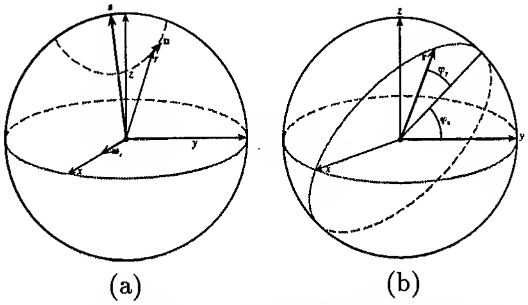


Figure 2: (a) Parameterization used in the analysis: $\boldsymbol{\omega}_\epsilon = \lambda(1,0,0)$, $\mathrm{s} = (0, \sin\chi, \cos\chi)$ with $\chi \in [0 \ldots \pi]$, $\mathbf{n} = \frac{\mathbf{s}\times\mathbf{r}}{|\mathbf{s}\times\mathbf{r}|}$. (b) Parameterization of $\mathbf{r}$ for normalization: φ_y is the angle between $\mathbf{r}$ and the yz plane; φ_x is the angle between the projection of $\mathbf{r}$ on the yz plane and some fiducial direction in the yz plane.

As shown in [Fermüller and Aloimonos, 1997], this can be achieved by multiplying the volume $V(\chi)$ for every direction χ by $\left|\frac{\sin\varphi_y}{\cos(\varphi_y)^2\cos(\chi-\varphi_x)^2-1}\right|$, where φ_x and φ_y are defined as described in Figure 2a.

Our focus is on the points in space with estimated negative range values $|\hat{\mathbf{R}}|$. Since $\mathbf{n} = \frac{\mathbf{s}\times\mathbf{r}}{|\mathbf{s}\times\mathbf{r}|}$ and $\mathbf{s}\cdot\boldsymbol{\omega}_\epsilon = 0$, we obtain from (4), by setting $f = 1$,

$$\left|\hat{\mathbf{R}}\right| = |\mathbf{R}| \frac{(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r}}{(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} - |\mathbf{R}|\,(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})\,(\mathbf{s}\cdot\mathbf{r})} < 0 \quad (7)$$

From this inequality the following constraint on $|\mathbf{R}|$ can be derived:

$$\begin{aligned}&\operatorname{sgn}(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r}\\ &\quad = -\operatorname{sgn}\left((\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} - |\mathbf{R}|(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r})\right) \quad (8)\end{aligned}$$

At any point $\mathbf{r}$ in the image this constraint is either satisfied for all values $|\mathbf{R}|$, or it is satisfied for an interval of values $|\mathbf{R}|$ bounded from either above or below, or it is not satisfied for any value at all. Thus, (7) provides a classification of the points on the sphere, and we obtain four different kinds of areas (types I–IV), as summarized in Table 1.

Table 1:

area	location	constraint on $\lvert\mathbf{R}\rvert$
I	$\operatorname{sgn}(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})$	$\lvert\mathbf{R}\rvert > \frac{(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r}}{(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})}$
II	$-\operatorname{sgn}(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})$	all $\lvert\mathbf{R}\rvert$
III	$\operatorname{sgn}(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} =$ $-\operatorname{sgn}(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})$	$\lvert\mathbf{R}\rvert < \frac{(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r}}{(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})}$
IV	$\operatorname{sgn}(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} =$ $\operatorname{sgn}(\hat{\mathbf{t}}\times\mathbf{s})\cdot\mathbf{r} =$ $-\operatorname{sgn}(\mathbf{r}\cdot\boldsymbol{\omega}_\epsilon)(\mathbf{r}\cdot\mathbf{s})$	none

Thus for any s, we obtain a volume of negative range values consisting of the volumes above areas I, II, and III. An illustration for both hemispheres is given in Figure 3. As can be seen, areas II and III cover the same amount of area, which has the size of the area between the two great circles $(\mathbf{t}\times\mathrm{s})\cdot\mathbf{r} = 0$ and $(\hat{\mathbf{t}}\times\mathrm{s})\cdot\mathbf{r} = 0$, and area I covers a hemisphere minus the area between $(\mathbf{t}\times\mathrm{s})\cdot\mathbf{r} = 0$ and $(\hat{\mathbf{t}}\times\mathrm{s})\cdot\mathbf{r} = 0$.

If the scene in view is unbounded, that is, $|\mathbf{R}| \in [0 \ldots \infty]$, there is a range of values $|\mathbf{R}|$ above any point $\mathbf{r}$ in areas I and III which results in negative range estimates. If we consider a lower bound $|\mathbf{R}_{\text{min}}| \neq 0$ and an upper bound $|\mathbf{R}_{\text{max}}| \neq \infty$, we obtain two additional curves C_{min} and C_{max} with $C_{\text{min}} = (\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} - |\mathbf{R}_{\text{min}}|\,(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r}) = 0$ and $C_{\text{max}} = (\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} - |\mathbf{R}_{\text{max}}|\,(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r}) = 0$ as bounds for areas with negative range values (as shown in Figure 3). As can be seen, the curves $C_{\text{min}} = 0$, $C_{\text{max}} = 0$, $(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r} = 0$ and $(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r}) = 0$ intersect at the same point.

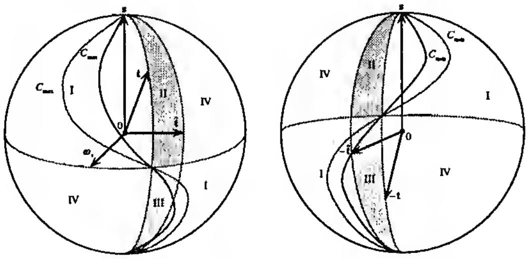


Figure 3: Classification of image points according to constraints on $|\mathbf{R}|$. At C_{min} and C_{max}, $|\mathbf{R}|$ is constrained to be greater (area I) or smaller (area III) than $|\mathbf{R}_{\text{min}}|$ or $|\mathbf{R}_{\text{max}}|$. The two hemispheres correspond to the front of the sphere and the back of the sphere, both as seen from the front of the sphere.

In area I, we do not obtain any volume of negative range estimates for points $\mathbf{r}$ between the curves $(\boldsymbol{\omega}_\epsilon \cdot \mathbf{r})(\mathbf{s} \cdot \mathbf{r}) = 0$ and $C_{\max} = 0$; the volume for points $\mathbf{r}$ between $C_{\min} = 0$ and $C_{\max} = 0$ is bounded from below by $|\mathbf{R}| = \frac{(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r}}{(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r})}$ (and from above by $|\mathbf{R}_{\max}|$), and the volume for points $\mathbf{r}$ between $C_{\min} = 0$ and $(\mathbf{t} \times \mathbf{s}) \cdot \mathbf{r} = 0$ extends from $|\mathbf{R}_{\min}|$ to $|\mathbf{R}_{\max}|$. In area III we do not obtain any volume for points $\mathbf{r}$ between $(\mathbf{t} \times \mathbf{s}) \cdot \mathbf{r} = 0$ and $C_{\min} = 0$. The volume for points $\mathbf{r}$ between $C_{\min} = 0$ and $C_{\max} = 0$ is bounded from above by $|\mathbf{R}| = \frac{(\mathbf{t}\times\mathbf{s})\cdot\mathbf{r}}{(\boldsymbol{\omega}_\epsilon\cdot\mathbf{r})(\mathbf{s}\cdot\mathbf{r})}$ (and from below by $|\mathbf{R}_{\min}|$) and the volume for points $\mathbf{r}$ between $C_{\max} = 0$ and $(\boldsymbol{\omega}_\epsilon \cdot \mathbf{r})(\mathbf{s} \cdot \mathbf{r}) = 0$ extends from $|\mathbf{R}_{\min}|$ to $|\mathbf{R}_{\max}|$.

We are given $\boldsymbol{\omega}_\epsilon$ and $\mathbf{t}$, and we are interested in $\hat{\mathbf{t}}$, which minimizes the negative range volume. For any $\mathbf{s}$ the negative range volume becomes smallest if $\hat{\mathbf{t}}$ is on the great circle of $\mathbf{t}$ and $\mathbf{s}$, that is, $(\mathbf{t} \times \mathbf{s}) \cdot \hat{\mathbf{t}} = 0$, as will be shown next.

Let us consider a $\hat{\mathbf{t}}$ such that $(\mathbf{t} \times \mathbf{s}) \cdot \hat{\mathbf{t}} \neq 0$, and let us change $\hat{\mathbf{t}}$ so that $(\mathbf{t} \times \mathbf{s}) \cdot \hat{\mathbf{t}} = 0$. As $\hat{\mathbf{t}}$ changes, the area of type II on the sphere becomes an area of type IV and the area of type III becomes an area of type I. Thus, the negative range volume obtained consists only of range values above areas of type I.

Let us use the following notation. A_{III-I} denotes the area which changes from type III to type I and V_{III} and $V_{I(III)}$ are the volumes before and after change. Similarly, A_{II-IV} denotes the area which changes from type II to type IV and V_{II} and V_{IV} are the corresponding volumes.

The change of $\hat{\mathbf{t}}$ does not have any effect on the volumes above the areas that did not change in type. However, the change of $\hat{\mathbf{t}}$ causes a decrease in the volume above the areas which changed in type: Volume $V_{I(III)} < V_{II}$. Furthermore, the normalization term is the same for points $\mathbf{r}_1(\varphi_{x_1}, \varphi_{y_1})$ and $\mathbf{r}_2(\varphi_{x_2}, \varphi_{y_2})$ symmetric with respect to the great circle $\mathbf{s} \cdot \mathbf{r} = 0$, because $\varphi_{y_1} = \varphi_{y_2}$ and $\varphi_{x_1} + \varphi_{x_2} = 2k\pi$ with $k \in \mathbb{N}$. Thus we encounter the same normalization factors in areas A_{III-I} and A_{II-IV}.

The volume of negative range values for any $\mathbf{s}$ is smallest for $(\mathbf{t} \times \mathbf{s}) \cdot \hat{\mathbf{t}} = 0$, independent of the range of values in which the scene lies. If we assume an upper bound $|\mathbf{R}_{\max}| \neq \infty$, or a lower bound $|\mathbf{R}_{\min}| \neq 0$, or both bounds, there exist points $\mathbf{r}$ in areas I and III above which there are no range values which contribute to the negative range volume. However, V_{II} is always larger than $V_{I(III)}$.

For any $\mathbf{s}$ the smallest volume is obtained for $\mathbf{s}$, $\mathbf{t}$, and $\hat{\mathbf{t}}$ lying on a great circle. Therefore, in order to minimize the total negative range volume, we must have $\mathbf{t} = \hat{\mathbf{t}}$.

Thus, in summary, we have shown that for any given rotational error $\boldsymbol{\omega}_\epsilon$ the negative range volume is smallest if the direction of the actual translation and the estimated translation coincide, that is, $\mathbf{t} = \hat{\mathbf{t}}$.

3.2 Fixed translational error

The analysis investigating the smallest rotational error, given a translational error, can be carried out in a way similar to the one above. For reasons of brevity only the idea is outlined here.

We are given $\mathbf{t}$ and $\hat{\mathbf{t}}$, and we are interested in the direction $\boldsymbol{\omega}_\epsilon$ minimizing the negative range volume. We choose the unit vectors $\mathbf{t}$, $\hat{\mathbf{t}}$, and $\mathbf{s}$ to lie in the xz plane.

We decompose $\boldsymbol{\omega}_\epsilon$ into a component $\boldsymbol{\omega}_{\text{par}}$ which lies in the xz plane and a component $\boldsymbol{\omega}_{\text{perp}}$ parallel to the y axis: $\boldsymbol{\omega}_\epsilon = \boldsymbol{\omega}_{\text{par}} + \boldsymbol{\omega}_{\text{perp}}$. It can be shown that if $\boldsymbol{\omega}_{\text{perp}} = 0$, the smallest negative range volume is obtained for $\boldsymbol{\omega}_{\text{par}_0} \neq 0$ parallel to the y axis with $(\mathbf{t} \times \boldsymbol{\omega}_{\text{par}_0}) = -(\hat{\mathbf{t}} \times \boldsymbol{\omega}_{\text{par}_0})$. A general $\boldsymbol{\omega}_\epsilon$ must satisfy the constraint $(\boldsymbol{\omega}_\epsilon \cdot \mathbf{t}) = (\boldsymbol{\omega}_\epsilon \cdot \hat{\mathbf{t}})$, but if we change the direction of $\boldsymbol{\omega}_\epsilon$, which amounts to $\frac{\boldsymbol{\omega}_{\text{par}}+\boldsymbol{\omega}_{\text{perp}}}{|\boldsymbol{\omega}_{\text{par}}+\boldsymbol{\omega}_{\text{perp}}|}$ with $\boldsymbol{\omega}_{\text{perp}} = \lambda(0, -1, 0)$, by continuously increasing $\lambda \geq 0$, the negative depth volume increases monotonically. Thus the smallest negative depth volume is obtained for $\boldsymbol{\omega}_{\text{par}_0}$, which lies on the geodesic through $\mathbf{t}$ and $\hat{\mathbf{t}}$ at an equal distance from both.

4 The planar case

If we use the more common component notation and express the coordinates of the focus of expansion as $(x_0, y_0) = (\frac{Uf}{W}, \frac{Vf}{W})$, and we set $W = 1$ and $\hat{W} = 1$, we obtain

$$\hat{Z} = Z \left(\frac{(x - \hat{x}_0)\, n_x + (y - \hat{y}_0)\, n_y}{\begin{array}{c}(x - x_0)\, n_x + (y - y_0)\, n_y \\ +Z\Big(\left(\alpha_\epsilon \frac{xy}{f} - \beta_\epsilon \left(\frac{x^2}{f} + f\right) + \gamma_\epsilon y\right) n_x \\ + \left(\alpha_\epsilon \left(\frac{y^2}{f} + f\right) - \beta_\epsilon \frac{xy}{f} - \gamma_\epsilon x\right) n_y \Big)\end{array}} \right) \tag{9}$$

where n_x and n_y denote the components of $\mathbf{n}$ in the x and y directions.

In the following analysis, we perform some simplification: For a limited field of view, the terms quadratic in the image coordinates, which appear in the rotational components, are small with respect to the linear and constant terms, and we therefore drop them.

The flow directions (n_x, n_y) can alternatively be written as $(\cos\psi, \sin\psi)$, with $\psi \in [0, \pi]$ denoting the angle between $[n_x, n_y]^T$ and the x axis.

To simplify the visualization of the volumes of negative depth in different directions, we perform the following coordinate transformation:

$$[x', y']^T = R[x, y]^T, [x_0', y_0']^T = R[x_0, y_0]^T$$
$$[\hat{x}_0', \hat{y}_0']^T = R[x, y]^T, [\alpha_\epsilon', \beta_\epsilon']^T = R[\alpha_\epsilon, \beta_\epsilon]^T$$

where $R = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}$.
The 0 distortion surface and the $-\infty$ distortion surface thus become

$$(x' - \hat{x}_0') = 0$$
$$\text{and} \quad (x' - x_0') + Z\left(-\beta_\epsilon' f + \gamma_\epsilon y\right) = 0$$

In the following proof we first consider the case of $\gamma_\epsilon = 0$ and then summarize the general case.

Part 1 ($\gamma_\epsilon = 0$)

If $\gamma_\epsilon = 0$, the volume of negative depth values for every direction ψ lies between the surfaces

$$(x' - \hat{x}_0') = 0 \quad \text{and} \quad (x' - x_0') - \beta_\epsilon' f Z = 0$$

The equation $(x' - \hat{x}_0') = 0$ describes a plane parallel to the $y'Z$ plane at distance $\hat{x}_0'$ from the origin, and the equation $(x' - x_0') - \beta_\epsilon' f Z = 0$ describes a plane parallel to the y' axis of slope $\frac{1}{\beta_\epsilon' f}$, which intersects the $x'y'$ plane at the x' coordinate x_0'. Thus we obtain a wedge-shaped volume parallel to the y' axis. Figure 4 illustrates the volume through a slice parallel to the $x'Z$ plane.

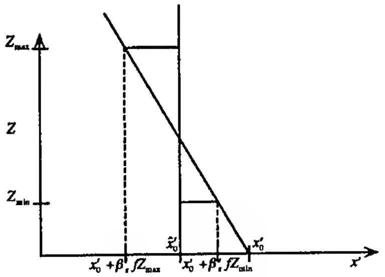

$Z_{\max}$
Z
$Z_{\min}$
$\hat{x}_0'$
x_0'
$x_0' + \beta_\epsilon' f Z_{\max}$
$x_0' + \beta_\epsilon' f Z_{\min}$
x'

Figure 4: Slice parallel to the $x'Z$ plane through the volume of negative estimated depth for a single direction.

Let us denote the area of this cross section by $A\psi$. If x_0' lies between $x_0' + \beta_\epsilon' f Z_{\min}$ and $x_0' + \beta_\epsilon' f Z_{\max}$,

$$A\psi = \left| x_{0_\epsilon}' (Z_{\max} + Z_{\min}) + \frac{\beta_\epsilon' f}{2}\left(Z_{\max}^2 + Z_{\min}^2\right) + \frac{x_{0_\epsilon}'}{\beta_\epsilon' f} \right| \qquad (10)$$

If we fix β_ϵ' and solve $\frac{\partial A\psi}{\partial x_{0_\epsilon}'} = 0$, we obtain $x_{0_\epsilon}' = -\frac{\beta_\epsilon' f}{2}(Z_{\max} + Z_{\min})$, that is, the 0 distortion surface has to intersect the $-\infty$ distortion surface in the middle of the depth interval in the plane $Z = \frac{Z_{\max}+Z_{\min}}{2}$.
Since $\beta_\epsilon' = \cos\psi\beta_\epsilon - \sin\psi\alpha_\epsilon$ and $x_{0_\epsilon} = \cos\psi x_{0_\epsilon} + \sin\psi y_{0_\epsilon}$, the volume is minimized for every direction if $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon}$. In other words, the rotational error $(\alpha_\epsilon, \beta_\epsilon)$ and the translational error $(x_{0_\epsilon}, y_{0_\epsilon})$ have to be perpendicular to each other. If, on the other hand, we fix x_{0_ϵ}', we obtain $x_{0_\epsilon}' = -\beta_\epsilon' f \frac{\sqrt{Z_{\max}^2 + Z_{\min}^2}}{2}$ and again $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon}$.

Part 2 ($\gamma_\epsilon \neq 0$)

If $\gamma_\epsilon \neq 0$, the $-\infty$ distortion surface becomes

$$(x' - \hat{x}_0') + Z\left(-\beta_\epsilon' f + \gamma_\epsilon y'\right) = 0$$

This surface can be most easily understood by slicing it with planes parallel to the $x'y'$ plane. At every depth value Z, we obtain a line of slope $\frac{-1}{\gamma_\epsilon Z}$ which intersects the x' axis in $x' = x_0' + \beta_\epsilon' f Z$ (see Figure 5).

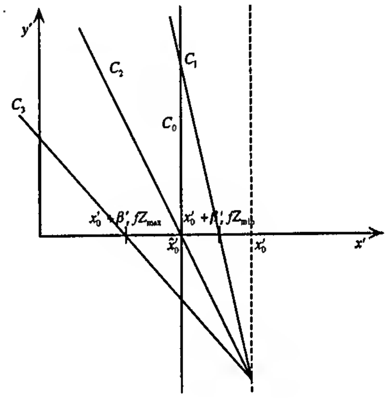

y'
C_2
C_1
C_3
C_0
$x_0' + \beta_\epsilon' f Z_{\max}$
$x_0' + \beta_\epsilon' f Z_{\min}$
$\hat{x}_0'$
x_0'
x'

Figure 5: Slices parallel to the $x'y'$ plane through the 0 distortion surface (C_0) and the $-\infty$ distortion surfaces at depth values $Z = Z_{\min}$ (C_1), $Z = -\frac{x_{0_\epsilon}'}{\beta_\epsilon' f}$ (C_2), and $Z = Z_{\max}$ (C_3).

In order to study the smallest negative depth volume for a given rotational error, we study how the volume changes as x_{0_ϵ}' changes from $x_0' + \beta_\epsilon' f \frac{(Z_{\max}+Z_{\min})}{2}$ to $x_0' + \beta_\epsilon' f \frac{(Z_{\max}+Z_{\min})}{2} + d$. As derived in [Fermüller and Aloimonos, 1997], we obtain for the smallest negative depth volume

$$d = \beta_\epsilon' f \left[\frac{(Z_{\max} - Z_{\min})}{\ln\left(\frac{Z_{\max}}{Z_{\min}}\right)} - \frac{1}{2}(Z_{\max} + Z_{\min}) \right]$$

and thus $\frac{x_{0_\epsilon}'}{\beta_\epsilon' f}$ depends only on the depth interval. Therefore we have the constraint $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon}$. For a given rotational error $(\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon)$, this constraint defines the direction of the FOE of the translational error on the image plane. For a given translational error $(x_{0_\epsilon}, y_{0_\epsilon})$ this constraint defines the direction of the AOR of the rotational error on the image. In addition we must have $\gamma_\epsilon = 0$.

Next let us fix only the amount of translational error $(x_{0_\epsilon}^2 + y_{0_\epsilon}^2)^{1/2}$. From the minimization of negative depth volume, we obtained $\beta_\epsilon{}' = \frac{x_{0_\epsilon}}{f}(\frac{2}{Z_{\max}^2 + Z_{\min}^2})^{1/2}$. Substituting for $\beta_\epsilon{}'$ into (10), we obtain $A\psi$ as a function of $x_{0_\epsilon}{}'$ and the depth interval. The negative depth volume for every direction ψ amounts to $A_\psi l_\psi$, where l_ψ denotes the average extent of the wedge-shaped negative depth volume in direction ψ, and the total negative depth volume is minimized if $\int_0^\pi A_\psi l_\psi d\psi$ is minimized. Considering a limited field of view, this is achieved if the actual and the estimated FOE lie on a line passing through the image center, that is, $\frac{x_0}{y_0} = \frac{\hat{x}_0}{\hat{y}_0}$.

5 Shape estimation in the presence of distortion

The above results are of great importance for the analysis of shape estimation. An error of the form $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon} = \frac{x_0}{y_0}$ guarantees that for the image near the fixation center, a shape map of the scene is derived which is very well behaved.

Near the image center the image coordinates are very small. Thus using (9) the distortion factor there can be approximated by

$$D = \frac{\hat{x}_0 n_x + \hat{y}_0 n_y}{x_0 n_x + y_0 n_y + Zf\,(\beta_\epsilon n_x - \alpha_\epsilon n_y)}$$

If $\frac{x_{0_\epsilon}}{y_{0_\epsilon}} = -\frac{\beta_\epsilon}{\alpha_\epsilon} = \frac{x_0}{y_0}$ for any given Z, the numerator is a multiple of the denominator and thus the distortion factor is the same for every direction (n_x, n_y). This means that scene points of the same depth are distorted by the same factor and the computed depth map has the same level contours as the actual depth map of the scene. All the distortion takes place only in the Z dimension. Thus the resulting depth function involves an affine transformation.

6 Conclusions

The inherent confounding of the translational and rotational parameters in the problem of reconstruction from multiple views has been analyzed. The results obtained, besides their potential use in structure-from-motion algorithms, also represent a computational analysis comparing different eye constructions in the natural world and different camera designs. The results on the sphere demonstrate that it is very easy for a system with panoramic vision to estimate its self-motion. Indeed, if the system possesses an inertial sensor that provides its rotation with some error, we have shown that after derotation, a simple algorithm considering only translation based on normal flow will estimate the translation optimally. This suggests that spherical eye design is optimal for flying systems such as the compound eyes of insects and the panoramic vision of birds.

The analysis on the plane reveals that for an optimal configuration of errors, the estimated depth distorts only in the z direction, with the level contours of the depth function distorting by the same amount, thus making it feasible to extract meaningful shape representations. This suggests that the camera-type eyes of primates, with high resolution near the center, are possibly optimal for systems that need good shape computation capabilities.

Acknowledgments

Special thanks to Sara Larson for her editorial and graphics assistance.

References

[Daniilidis and Spetsakis, 1996] K. Daniilidis and M.E. Spetsakis. Understanding noise sensitivity in structure from motion. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Chapter 4. Lawrence Erlbaum Associates, Hillsdale, NJ, 1996.

[Faugeras, 1992] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1992.

[Fermüller and Aloimonos, 1997] C. Fermüller and Y. Aloimonos. Algorithm-independent stability analysis of structure from motion. *International Journal of Computer Vision*, 1997. In press.

[Hartley, 1994] R.I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1036–1041, 1994.

[Horn and Weldon, Jr., 1988] B.K.P. Horn and E.J. Weldon, Jr. Direct method for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[Maybank, 1993] S. Maybank. *Theory of Reconstruction from Image Motion.* Springer, Berlin, 1993.

Reflectance and Texture of Real-World Surfaces

Kristin J. Dana
Shree K. Nayar
Department of Computer Science
Columbia University
New York, NY 10027
Email: *dana@cs.columbia.edu*
nayar@cs.columbia.edu

Bram van Ginneken
Jan J. Koenderink
Department of Physics
Utrecht University
3508 TA Utrecht, the Netherlands
Email: *b.vanginneken@fys.ruu.nl*
j.j.koenderink@fys.ruu.nl

Abstract

In this work, we investigate the visual appearance of real-world surfaces and the dependence of appearance on imaging conditions. We present a BRDF (bidirectional reflectance distribution function) database with reflectance measurements for over 60 different samples, each observed with over 200 different combinations of viewing and source directions. We fit the BRDF measurements to two recent models to obtain a BRDF parameter database. These BRDF parameters can be directly used for both image analysis and image synthesis. Finally, we present a BTF (bidirectional texture function) database with image textures from over 60 different samples, each observed with over 200 different combinations of viewing and source directions. Each of these unique databases has important implications for a variety of vision algorithms and each is made publicly available.

1 Introduction

Characterizing the appearance of real-world surfaces is important for many computer vision algorithms. The appearance of any surface is a function of the scale at which it is observed. When the characteristic variations of the surface are subpixel, all local image pixels have the same intensity determined by the surface *reflectance.* The variation of radiance with viewing and illumination direction is captured by the BRDF (*bidirectional reflectance distribution function*). If the characteristic surface undulations are instead projected onto several image pixels, there is a local variation of pixel intensity, referred to as image *texture.* The dependency of texture on viewing and illumination directions is described by the BTF (*bidirectional texture function*). This taxonomy is illustrated in Figure 1.

In this work we measure the BRDF of over 60 samples of rough, real-world surfaces. Although BRDF models have been widely discussed and used in vision (see [10],[16],[19],[7],[12]) the BRDFs of a large and diverse collection of real-world surfaces have never before been obtained. Our measurements comprise a comprehensive BRDF database (the first of its kind) that is now publicly available at *www.cs.columbia.edu/CAVE/curet.* Exactly how well the BRDFs of real-world surfaces fit existing models has remained unknown as each model is typically verified using a small number (2 to 6) of surfaces. Our large database allows us to evaluate the performance of known models. Specifically, the measurements are fit to two existing analytical representations: the Oren-Nayar model [12] for surfaces with isotropic roughness and the Koenderink et al. decomposition [7] for both anisotropic and isotropic surfaces. Our fitting results form a concise BRDF parameter database that is also publicly available at *www.cs.columbia.edu/CAVE/curet.* These BRDF parameters can be directly used for both image analysis and image synthesis. In addition, the BRDF measurements can be used to evaluate other existing models [10],[16],[19] as well as future models.

While obtaining BRDF measurements, images of each real-world sample are recorded. These images prove valuable since they comprise a texture database, or a BTF database, with over 12,000 images (61 samples with 205 images per sample). Current literature deals almost exclusively with textures due to albedo and color variations on planar surfaces (see [18],[2],[6]). In contrast, the texture due to surface roughness has complex dependencies on viewing and illumination directions. These dependencies cannot be studied using existing tex-

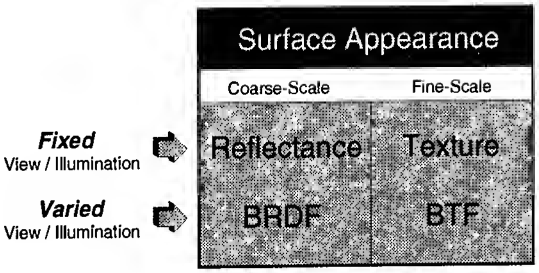


Figure 1: Taxonomy of surface appearance. When viewing and illumination directions are fixed, surface appearance can be described by either *radiance* (at coarse-scale observation) or *texture* (at fine-scale observation). When viewing and illumination directions vary, the equivalent descriptions are the *bidirectional reflectance distribution function* (BRDF) and the *bidirectional texture function* (BTF).

ture databases that include few images (often a single image) of each sample (for instance, the widely used the Brodatz database). Our texture database covers a diverse collection of rough surfaces and captures the variation of image texture with changing illumination and viewing directions. This database is also available at *www.cs.columbia.edu/CAVE/curet.*

The measurements and model fitting results are pertinent to a variety of areas including remote-sensing, photogrammetry, image understanding and scene rendering. Important implications of this work for computer vision are discussed.

2 Measurement Methods

Our measurement device is comprised of a robot[1], lamp[2], personal computer[3], spectrometer[4] and video camera[5]. Measuring the BRDF requires radiance measurements for a range of viewing/illumination directions. For each sample and each combination of illumination and viewing directions, an image from the video camera is captured by the frame grabber. These images have 640x480 pixels with 24 bits per pixel (8 bits per R/G/B channel). The pixel values are converted to radiance values using a post-processing calibration and segmentation scheme described in [3]. The calibrated, segmented images serve as the BTF measurements and these images are averaged to obtain the BRDF measurements.

The need to vary the viewing and source directions over the entire hemisphere of possible directions presents a practical obstacle in the measurements. This difficulty is reduced considerably by orienting the sample to generate the varied conditions. As illustrated in Figure 2, the light source remains fixed throughout the measurements. The light rays incident on the sample are approximately parallel and uniformly illuminate the sample. The camera is mounted on a tripod and its optical axis is parallel to the floor of the lab. During measurements for a given sample, the camera is moved to seven different locations, each separated by 22.5 degrees in the ground plane at a distance of 200 cm from the sample. For each camera position, the sample is oriented so that its normal is directed toward the vertices of the facets which tessellate the fixed quarter-sphere illustrated in Figure 2. With this arrangement, a considerable number of measurements are made in the plane of incidence (i.e. source direction, viewing direction and sample normal lie in the same plane). Also, for each camera position, a specular point is included where the sample normal bisects the angle between the viewing and source direction. Sample orientations with corresponding viewing angles or illumination angles greater than 85 degrees are excluded from the measurements to avoid self-occlusion and self-shadowing. This exclusion results in the collection of 205 images for each sample. For anisotropic samples, the 205 measurements are repeated after rotating the sample about the global normal by either 90 degrees or 45 degrees, depending on the structure of the anisotropy.

3 Samples For Measurements

The collection of real-world surfaces used in the measurements are illustrated in Figure 3. Samples of these surfaces were mounted on 10x12 cm bases which were constructed to fit onto the robot gripper. Each sample, though globally planar, exhibits considerable depth variation or macroscopic surface roughness. The samples were chosen to span a wide range of geometric and photometric properties. The categories include specular surfaces (aluminum foil, artificial grass), diffuse surfaces (plaster, concrete), isotropic surfaces (cork, leather, styrofoam), anisotropic surfaces (straw, corduroy, corn husk), surfaces with large height variations (crumpled paper, terrycloth, pebbles), surfaces with small height variations (sandpa-

[1] SCORBOT-ER V by ESHED Robotec (Tel Aviv, Israel).

[2] Halogen bulb with a Fresnel lens.

[3] IBM compatible PC running Windows 3.1 with "Videomaker"
frame grabber by VITEC Multimedia.

[4] SpectraScan PR-704 by Photoresearch (Chatsworth,CA).

[5] Sony DXC-930 3-CCD color video camera.

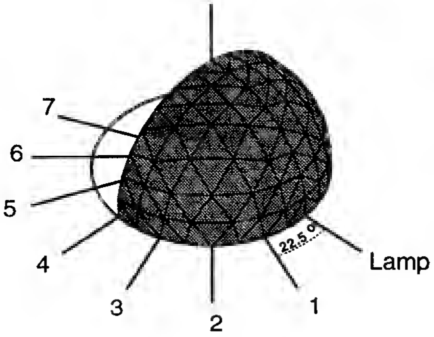


Figure 2: Illustration of the discrete sample orientations, light source and camera positions used in the measurements. For each of the 7 camera positions illustrated, the robot orients the sample's global normal to the directions indicated by the vertices on the quarter-sphere. The illumination direction remains fixed.

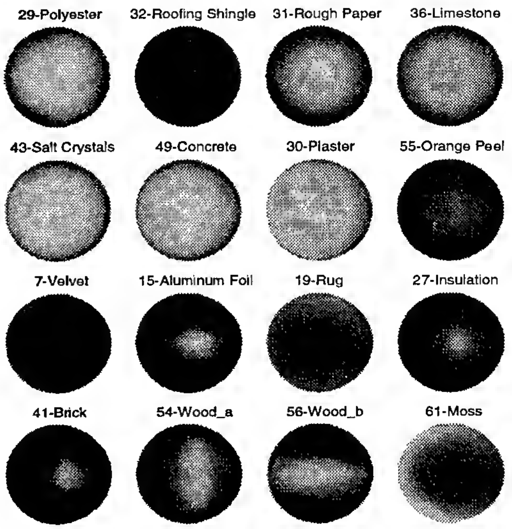


Figure 4: Spheres rendered using the BRDF measurements obtained from camera position 1 (illumination at 22.5° to the right). Interpolation was used to obtain radiance values between the measured points.

per, quarry tile, brick), pastel surfaces (paper, cotton), colored surfaces (velvet, rug), natural surfaces (moss, lettuce, fur) and man-made surfaces (sponge, terrycloth, velvet).

4 BRDF Database

The BRDF measurements form a database with over 12,000 reflectance measurements (61 samples, 205 measurements per sample, 205 additional measurements for anisotropic samples). The measured BRDFs are quite diverse and reveal the complex appearance of many ordinary surfaces.

Figure 4 illustrates examples of spheres rendered with the measured BRDF as seen from camera position 1, i.e. with illumination from 22.5° to the right. Interpolation is used to obtain a continuous radiance pattern over each sphere. The rendered sphere corresponding to velvet (Sample 7) shows a particularly interesting BRDF that has bright regions when the global surface normal is close to 90 degrees from the illumination direction. This effect can be accounted for by considering the individual strands comprising the velvet structure which reflect light strongly as the illumination becomes oblique. This effect is consistent with the observed brightness in the interiors of folds of a velvet sheet. Indeed, the rendered velvet sphere gives a convincing impression of velvet.

The rendered spheres of plaster (Sample 30) and roofing shingle (Sample 32) show a fairly flat appearance which is quite different from the Lambertian prediction for such matte objects, but is consistent with [11] and [12]. Concrete (Sample 49) and salt crystals (Sample 43) also show a somewhat flat appearance, while rough paper (Sample 31) is more Lambertian. The plush rug (Sample 19) and moss (Sample 61), have similar reflectance patterns as one would expect from the similarities of their geometry. Rendered spheres from two anisotropic samples of wood (Sample 54 and Sample 56) are also illustrated in Figure 4. The structure of the anisotropy of sample 54 consists of horizontally oriented ridges. This ridge structure causes a vertical bright stripe instead of a specular lobe in the rendered sphere. Sample 56 shows a similar effect, but the anisotropic structure for this sample consists of near vertical ridges. Consequently the corresponding rendered sphere shows a horizontal bright region due to the surface geometry.

5 Fitting to BRDF Models

A concise description is required for functional utility of the measurements. We employ the Oren-Nayar model [12] and the Koenderink et al. representation [8] to obtain parametric descriptions of the BRDF measurement database. The resulting database of parameters can be used directly and conveniently in a variety of algorithms where accurate, concise and analytical reflectance descriptions are needed. In vision, these applications include shape-from-shading and photometric stereo. In computer graphics, the reflectance parameters are useful for realistic rendering of natural surfaces. As with the

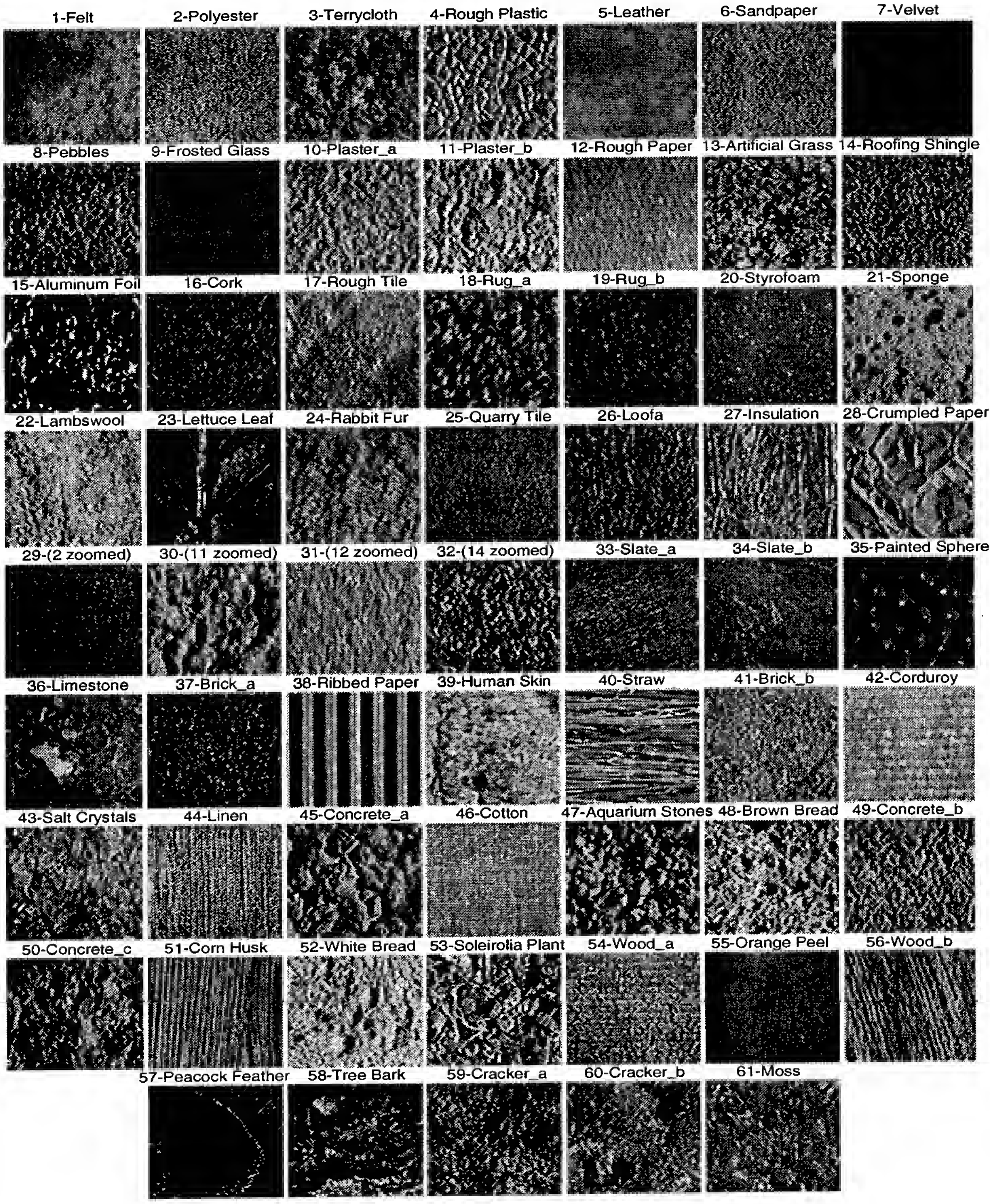


Figure 3: The collection of 61 real-world surfaces used in the measurements. The name and number of each sample is indicated above its image. The samples were chosen to span a wide range of geometric and photometric properties. The categories include specular surfaces (aluminum foil, artificial grass), diffuse surfaces (plaster, concrete), isotropic surfaces (cork, leather, styrofoam), anisotropic surfaces (straw, corduroy, corn husk), surfaces with large height variations (crumpled paper, terrycloth, pebbles), surfaces with small height variations (sandpaper, quarry tile, brick), pastel surfaces (paper, cotton), colored surfaces (velvet, rug), natural surfaces (moss, lettuce, fur) and man-made surfaces (sponge, terrycloth, velvet). Different samples of the same type of surfaces are denoted by letters, e.g. Brick_a and Brick_b. Samples 29, 30, 31 and 32 are close-up views of samples 2, 11, 12 and 14, respectively.

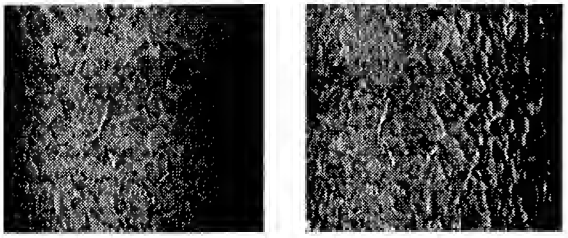


Figure 5: Cylinder rendered with 2D texture mapping (left) and 3D texture mapping (right). The 2D texture mapping was done by warping a frontal view image of the texture (with illumination at 22.5 degrees to the right). The 3D texture mapping uses 13 images from the BTF of Sample 45 (concrete).

measurement database, the complete database of reflectance parameters is also available electronically.

6 Texture Database

The appearance of a rough surface, whether manifested as a single radiance value or as image texture, depends on viewing and source direction. Just as the BRDF describes the coarse-scale appearance of a rough surface, the BTF (bidirectional texture function) is useful for describing the fine-scale appearance of a rough surface. Our measurements of image texture comprise the first BTF database for real-world surfaces. The database has over 12,000 images (61 samples, 205 measurements per sample, 205 additional measurements for anisotropic samples).

Important observations on the BTF can be made from the database. Consider texture mapping using Sample 45 (concrete) as shown in Figure 5. The differences in the 2D texture-mapped cylinder and the 3D texture-mapped cylinder (using the database images) are readily apparent. Because of the varying surface normals across the sample, foreshortening effects are quite complicated and cannot be accounted for by common texture-mapping techniques. A detailed discussion of the pitfalls of current texture rendering schemes is given in [8].

Consider the same sample shown under two different sets of illumination and viewing directions in Figure 6. The corresponding Fourier spectra are also shown in Figure 6. Notice that the spectra are quite different. Most of the difference is due to the change in azimuthal angle of the source direction which causes a change in the shadowing direction and hence a change in the dominant orientation of the spectrum. If the image texture was due to a planar albedo or color variation, changes in the source direction would not have this type of effect on the spectrum. Source direction changes would only cause a uniform scaling of the intensity over the entire image.

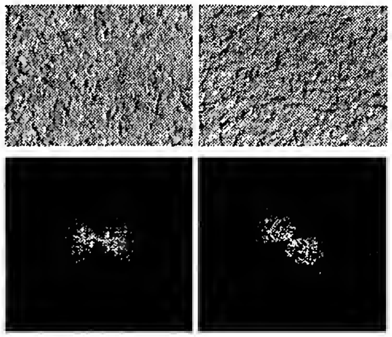


Figure 6: Changes in the spectrum due to changes in imaging conditions. (Top row) Two images of sample 11 with different source and viewing directions. (Bottom row) Fourier spectrum of the images in the top row, with zero frequency at the center and brighter regions corresponding to higher magnitudes. The orientation change in the spectrum is due to the change of source direction which causes a change in the shadow direction.

7 Implications for Vision

Our BRDF measurement database provides a thorough investigation of the reflectance properties of real-world rough surfaces. This database fills a long-standing need for a benchmark to test and compare BRDF models as we have done here for the Oren-Nayar model and the Koenderink et al. decomposition.

Our BRDF parameter database, obtained by fitting the measurements to the Oren-Nayar model and the Koenderink et al. decomposition, can be used in place of the popular Lambertian reflectance model in such algorithms as shape-from-shading [5] and photometric stereo [20]. Since these algorithms rely on a reflectance model to ascertain shape, inaccuracies of the Lambertian model can significantly affect their performance. The model parameters can also be used instead of popular shading models [4],[1] for photorealistic rendering of real-world surfaces.

Since the parameter database covers two BRDF representations, a choice can be made to balance accuracy and conciseness. For isotropic surfaces the 3-parameter Oren-Nayar model can be employed. For isotropic and anisotropic surfaces, when a richer description can be afforded, the 55 parameter Koenderink et al. model can be used. For the 61 surfaces we have investigated, the parameters for both models are readily available. Our BTF database is the first comprehensive investigation of texture appearance as a function of viewing and illumination direction. As illustrated in Figure 5 and Figure 6, surface roughness causes notable effects on the BTF which are not considered by current texture algorithms. Present algorithms for shape-from-texture [13],[15],[9], texture segmentation [21],[9] and texture recognition [14] are only suitable for 2D textures, i.e. planar texture due to albedo variation. Texture rendering also typically assumes a 2D planar texture that is mapped to a 3D surface. When the surface is rough, the rendering tends to be too flat and unrealistic. Texture analysis and synthesis of real-world rough surfaces remains an important unsolved problem. The database illustrates the need for 3D texture algorithms and serves as a starting point for their exploration.

Our BRDF measurement database, BRDF model parameter database and BTF measurement database together represent an extensive investigation of the appearance of real-world surfaces. Each of these databases has important implications for computer vision.

Acknowledgements

The research is sponsored in part by the National Science Foundation, DARPA/ONR under the MURI Grant No. N00014-95-1-0601 and by REALISE of the European Commission.

References

[1] P. Bui-Tuong, "Illumination for Computer Generated Pictures," *Communications of the ACM*, Vol. 18, pp. 311-317, 1975.

[2] S. Chatterjee, "Classification of Natural Textures using Gaussian Markov Random Fields," Markov Random Fields: Theory and Applications, pp. 159-177, Academic Press, Boston, 1993.

[3] K. J. Dana, B. Van Ginneken, S. K. Nayar and J.J. Koenderink, *Columbia University Technical Report CUCS-048-96*, December 1996.

[4] H. Gouraud, "Continous Shading of Curves Surfaces," *IEEE Trans. on Computers*, pp. 623-629, June 1971.

[5] B.K.P. Horn and M.J. Brooks, *Shape from Shading*, MIT Press, Cambridge, Mass, 1989.

[6] R.L. Kashyap, "Characterization and Estimation of Two-Dimensional ARMA Models," *IEEE Transactions on Information Theory*, Vol. IT-30, No. 5, September 1984.

[7] J.J. Koenderink, A.J. van Doorn and M. Stavridi, "Bidirectional reflection distribution function expressed in terms of surface scattering modes," *European Conference on Computer Vision* , pp. 28-39, 1996.

[8] J.J. Koenderink and A.J. van Doorn, "Illuminance texture due to surface mesostructure," *Journal of the Optical Society of America A*, Vol. 13 pp. 452-463, 1996.

[9] J. Krumm and S.A. Shafer, "Texture segmentation and shape in the same image," *IEEE Conference on Computer Vision*, pp. 121-127, 1995.

[10] S.K. Nayar, K. Ikeuchi and T. Kanade, "Surface Reflection: Physical and Geometrical Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 611-634, July 1991.

[11] S.K. Nayar and M. Oren, "Visual appearance of matte surfaces," Science, Vol. 267, pp. 1153-1156, Feb. 1995.

[12] M. Oren and S.K. Nayar, "Generalization of the Lambertian model and implications for machine vision," *International Journal of Computer Vision* , Vol. 14, pp. 227-251, 1995.

[13] M.A.S. Patel and F.S. Cohen, "Shape from texture using Markov random field models and stereo-windows," *IEEE Conference on CVPR*, pp. 290-305, 1992.

[14] R.W. Picard, T. Kabir and F. Liu, " Real-time recognition with the entire Brodatz texture database," *IEEE Conference on CVPR*, pp. 638-9, 1993.

[15] B.J. Super and A.C. Bovik, "Shape from texture using local spectral moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 333-343, 1995.

[16] H.D. Tagare and R.J.P. DeFigueiredo, "A Framework for the Construction of Reflectance Maps for Machine Vision," *CVGIP: Image Understanding*, Vol. 57, No. 3, pp. 265-282, May 1993.

[17] K.E. Torrance and E.M. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces," *Journal of the Optical Society of America*, Vol. 57, No. 9, pp. 1105-1114, 1967.

[18] L. Wang and G. Healey, "Illumination and geometry invariant recognition of texture in color images," *IEEE Conference on CVPR*, pp. 419-424, 1996.

[19] L.B. Wolff, "A Diffuse Reflectance Model for Smooth Dielectrics," *Journal of the Optical Society of America A - Special Issue on Physics Based Machine Vision*, Vol. 11, pp. 2956-2968, November 1994.

[20] R.J. Woodham, "Photometric Methods for Determining Surface Orientation from Multiple Images," *Optics Engineering*, Vol. 19, No. 1, pp. 139-144, 1980.

[21] Z. Xie and M. Brady, " Texture segmentation using local energy in wavelet scale space," *ECCV*, Vol. 1, pp. 304-313, 1996.

Parametric Feature Detection*

Simon Baker†, Shree K. Nayar†, and Hiroshi Murase‡
†Department of Computer Science, Columbia University, New York, USA
‡NTT Basic Research Laboratory, Atsugi-shi, Kanagawa, Japan

Abstract

We propose an algorithm to automatically construct feature detectors for arbitrary parametric features. In the algorithm, each feature is represented as a densely sampled parametric manifold in a low dimensional subspace of $\Re^N$. Detection is performed by projecting the brightness distribution around each image pixel into the subspace. If the projection lies sufficiently close to the feature manifold, the feature is detected and the location of the closest point on the manifold is used to estimate the feature parameters. By applying the algorithm to appropriate feature models, detectors have been constructed for five parametric features, namely, step edge, roof edge, line, corner, and circular disc.

1 Introduction

Many applications in computational vision rely upon robust detection of image features and accurate estimation of their parameters. Although the standard example of such a feature is the *step edge,* it is by no means the only feature of interest. A comprehensive list would also include *lines, corners, junctions,* and *roof edges*[1] as well as numerous others. In short, features may be too numerous to justify the process of deriving a new detector for each one. Our aim in this paper is to develop a single detection mechanism that can be applied to any parametric feature. Moreover, we wish to obtain precise estimates of feature parameters, which if recovered with precision can be of vital importance to higher levels of visual processing.

To obtain high performance in both feature detection and parameter estimation, it is essential to accurately model the features as they appear in the physical world. Hence, we choose not to make any simplifying assumptions for analytic or efficiency reasons, and instead use realistic multiparameter feature models. Further, we give careful consideration to the conversion of the continuous radiance function of the feature in the world to its discrete image.

Given a parametric model of a feature and a model of the imaging system, we can accurately predict the pixel brightness values in a window about an imaged feature. If we regard the pixel brightness values as real numbers, we can treat each feature as corresponding to a parametric manifold in $\Re^N$, where N is the number of pixels in the window surrounding the feature. Feature detection is then posed as finding the closest point on the manifold to the point in $\Re^N$ corresponding to the pixel brightness values in a novel image window. If the closest manifold point is near enough to the novel point, we detect the feature and the exact location (parameters) of the closest manifold point may be used as estimates of the parameters of the feature. This statement of the feature detection problem was first introduced by Hueckel [1971] and was subsequently used by Hummel [1979] amongst others.

Hueckel and Hummel both argued that, in order to achieve high efficiency, a closed form solution must be found for (the parameters of) the closest manifold point. To make their derivations possible they used simplified feature models. Our view of feature detection is radically different. We argue that the features we wish to detect are inherently complex visual entities and so give up all hope of finding closed-form solutions for the best-fit parameters. Instead, we discretize the search problem by densely sampling the feature manifold.

At first glance, finding the closest sample point may seem inefficient to the point of impracticality. However, we will demonstrate that our approach is very practical through a combination of normalization, dimension reduction [Nayar *et al.*, 1996], efficient heuristic search [Baker *et al.*,

*This research was supported in parts by ARPA Contract DACA-76-92-C-007, DOD/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and the NTT Basic Research Laboratory.

[1] Given the extent to which feature detection has been explored, a survey of the work in this area is well beyond the scope of this paper. In our discussion, we only use examples of previous detectors without attempting to mention all of them. Further, we will primarily be interested in examples that use parametric feature models rather than those based upon differential invariants.

1998], and rejection techniques [Baker and Nayar, 1996b]. Even in the present unoptimized implementation, feature detection and parameter estimation take only a few seconds on a standard single-processor workstation when applied to a 512 × 480 image.

2 Parametric Feature Representation

2.1 Parametric Scene Features

By a scene feature we mean a geometric or photometric phenomenon that produces spatial radiance variations which can aid in visual perception. The continuous radiance function of the scene feature can be written as $F^c(x, y; \mathbf{q})$ where $(x, y) \in S$ are points within a feature window S and $\mathbf{q}$ are the parameters of the feature.

2.2 Image Formation and Sensing

Previous work on feature detection has implicitly assumed that artifacts induced by the imaging system are negligible and can be ignored. We make our models as precise as possible by incorporating these effects. One such effect is defocus. Another is that the finite size of the lens aperture causes the optical transfer function to be spatially bandlimited. Also, the feature itself, even before imaging, may be somewhat smoothed or rounded. The defocus factor can be approximated as a pillbox function [Born and Wolf, 1965], the optical transfer function by the square of the first-order Bessel function of the first kind [Born and Wolf, 1965], and the blurring due to imperfections in the feature by a Gaussian function [Koenderink, 1984]. We combine all three effects into a single blurring factor that is assumed to be a 2-D Gaussian function:

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{1}{2} \cdot \frac{x^2 + y^2}{\sigma^2}) \qquad (1)$$

The continuous image on the sensor plane is converted to a discrete image through two processes. First, the light flux falling within each pixel is integrated. If the pixels are rectangular in structure [Barbe, 1980] [Norton, 1982], the averaging function is:

$$a(x, y) = \frac{1}{w_x\, w_y} {}^2\Pi(\frac{1}{w_x} x, \frac{1}{w_y} y) \qquad (2)$$

where w_x and w_y are the dimensions of the pixel. Next, the pixels are sampled, which we model by the rectangular grid:

$$s(x, y) = {}^2\mathrm{III}(\tfrac{1}{p_x} x, \tfrac{1}{p_y} y) \qquad (3)$$

where p_x and p_y are the spacings between samples. The final discrete image of a feature may then be written as: $F(x, y; \mathbf{q}) =$

$$\{\, F^c(x, y; \mathbf{q}) * g(x, y) * a(x, y) \,\} \cdot s(x, y) \qquad (4)$$

where $*$ is the 2-D convolution operator. Since the above is a weighted sum of Dirac delta functions, it can be rewritten as $F(m, n; \mathbf{q})$, where $(m, n) \in S$ are the (integral) pixel coordinates.

2.3 Parametric Feature Manifolds

If the number of pixels (m, n) in the window S is N, each feature instance $F(m, n; \mathbf{q})$ may be regarded as a point in $\Re^N$. Suppose the feature has k parameters: dim($\mathbf{q}$)=k. Then, as the parameters vary over their ranges, $F(m, n; \mathbf{q})$ traces out a k-parameter manifold. Feature detection is then posed as finding the closest point on the feature manifold to the point in $\Re^N$ corresponding to each window in the image. If the manifold is near enough, we detect the feature and the location (parameters) of the closest manifold point provides an estimate of the feature parameters.

2.4 Parameter Normalization

For each feature instance $F(m, n; \mathbf{q})$ encountered, we compute its mean pixel value $\mu(\mathbf{q}) = \frac{1}{N} \sum_{(n,m)\in S} F(m, n; \mathbf{q})$, and its pixel variance $\nu(\mathbf{q}) = \|\, F(m, n; \mathbf{q}) - \mu(\mathbf{q}) \,\|$. We then apply the following brightness normalization:

$$\overline{F}(m, n; \mathbf{q}) = \frac{1}{\nu(\mathbf{q})} [F(m, n; \mathbf{q}) \; - \; \mu(\mathbf{q})] \qquad (5)$$

For all of the features we have considered, the above normalization reduces the dimensionality of the feature manifold by two. This happens because $\overline{F}(m, n; \mathbf{q})$ is (approximately) independent of two of the parameters in $\mathbf{q}$. Once a feature has been detected, μ and ν can be used to recover the two normalized parameters [Baker *et al.*, 1998].

2.5 Dimension Reduction

For several reasons, such as feature symmetries and high correlation between feature instances with similar parameter values, it is possible to represent the feature manifold in a low-dimensional subspace of $\Re^N$ without significant loss of information[2]. If correlation between fea-

[2] This idea was first explored in [Hummel, 1979]. Whereas Hummel derived closed-form solutions based upon simplistic feature models, our approach is to use elaborate feature models and numerical methods. This results in higher precision and greater generality. A similar approach has been adopted in [Nandy *et al.*, 1996].

ture instances is the preferred measure of similarity, the Karhunen-Loéve (K-L) expansion [Fukunaga, 1990], yields the optimal subspace.

3 Example Features

For lack of space, we now illustrate the parametric manifold representations for only 1 of the 5 features which we constructed detectors for. The results for the other features are similar and may be found in [Baker *et al.*, 1998].

3.1 Step Edge

Figures 1(a) and 1(b) show isometric and plan views of our step edge model. It is a generalization of the models used in [Hueckel, 1971], [Hummel, 1979], and [Lenz, 1987]. It is particularly similar to the model of [Nalwa and Binford, 1986], differing only slightly in its treatment of smoothing effects.

The basis for the 2-D step edge model is the 1-D unit step function:

$$u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases} \qquad (6)$$

A step with lower intensity level A and upper intensity level $A{+}B$ can be written as $A{+}B{\cdot}u(t)$. To extend to 2-D, we assume that the step edge is of constant cross section, is oriented at angle θ to the x-axis, and lies at distance ρ from the origin. Then, the perpendicular distance of an arbitrary 2-D point (x, y) from the step is given by:

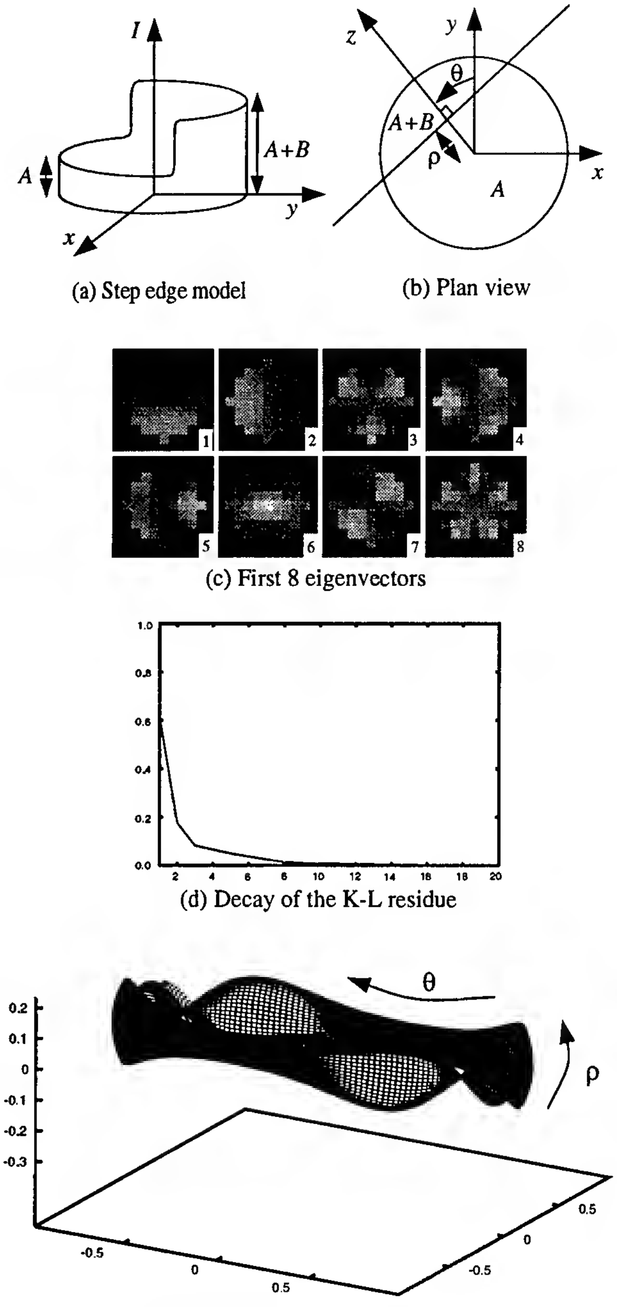
$$z = y \cdot \cos\theta - x \cdot \sin\theta - \rho \qquad (7)$$

Therefore, an ideal step edge of arbitrary orientation and displacement from the origin is given by the 2-D function $A + B \cdot u(z)$. For the reasons given in Section 2.2 we incorporate the Gaussian blurring function, the pixel averaging function, and the sampling function. Finally, the step edge model is: $F_{SE}(x, y; A, B, \theta, \rho, \sigma) =$

$$\{\,(A{+}B{\cdot}u(z)){*}g(x, y; \sigma){*}\ a(x, y)\,\}\,.\,s(x, y) \quad (8)$$

where z is given by Equation (7).

The step edge model has 5 parameters, namely, orientation θ, localization ρ, blurring or scaling σ, and the brightness values A and B. The orientation parameter θ is drawn from $[0^o, 360^o]$. We restrict the localization parameter ρ to lie in $[-1/\sqrt{2}, 1/\sqrt{2}]$, since any edge must pass closer than $1/\sqrt{2}$ pixels from the center of at least



(a) Step edge model

(b) Plan view

(c) First 8 eigenvectors

(d) Decay of the K-L residue

(e) Step edge parametric manifold

Figure 1: The step edge model includes two constant intensity regions of brightness A and $A{+}B$. Its orientation and intrapixel displacement from the origin are given by the parameters θ and ρ respectively. The fifth parameter (not shown) is the blurring factor σ. The K-L residue plot shows that 90% of the edge image content is preserved by the first 3 eigenvectors. The step edge manifold is parameterized by orientation and intrapixel localization for a fixed blurring value and is displayed in a 3-D subspace.

one pixel in the image. The blurring parameter $\sigma \in [0.3, 1.5]$. The intensity parameters A and B are free to take any value because of the normalization described in Section 2.4. The structure of a normalized step edge is independent of A & B and is uniquely determined by the parameters θ, ρ, and σ. Further, the values of A and B may be recovered from the values of μ and ν calculated during normalization [Baker *et al.*, 1998].

The window chosen for our edge model is a 49 pixel disc to avoid unnecessary non-linearities induced by a square window. The results of applying the Karhunen-Loéve expansion are displayed in Figures 1(c) and 1(d). In Figure 1(c) we display the 8 most important eigenvectors, ranked by their eigenvalues. The similarity between the first 4 eigenvectors and the ones derived in [Hummel, 1979] is immediate. On closer inspection, however, we notice that while Hummel's eigenvectors are radially symmetric, the ones we computed are not. This is to be expected since the introduction of the parameters ρ and σ breaks the radial symmetry in Hummel's edge model.

In Figure 1(d), the decay of the Karhunen-Loéve residue (sum of eigenvalues discarded) is plotted as a function of the number of eigenvectors. To reduce the residue to 10% we need to use 3 eigenvectors. To reduce it further to 2% we need 8 eigenvectors. Figure 1(d) illustrates a significant data compression factor of 5-15 times. As a result, feature detection is made far more efficient.

The step edge manifold is displayed in Figure 1(e). Naturally, we are only able to display a projection of it into a 3-D subspace. This subspace is the one spanned by the 3 most important eigenvectors. For clarity, we only display a 2 parameter slice through the manifold, obtained by keeping σ constant while varying θ and ρ.

4 Feature Detection

Given a point in $\Re^N$ corresponding to the pixel intensity values in a novel feature window, feature detection requires finding the closest point on the parametric manifold. If the distance between the novel point and the closest manifold point is sufficiently small, we declare the presence of the feature. The parameters of the closest manifold point are then used as estimates of the scene feature's parameters. If the distance between the novel point and the manifold is too large, we assert the absence of the feature.

We approximate the closest manifold point by densely sampling the manifold and then performing a search for the closest sample point. So long as we sample densely enough, this yields a sufficiently good estimate of the closest manifold point. We search using a heuristic coarse-to-fine search which takes advantage of the relatively smooth manifolds [Baker *et al.*, 1998].

As an example of the search complexity for the step edge model, if we sample θ every 1.6°, ρ every 0.088 pixel, and σ every 0.14 pixel, we have 46,368 sample points. Then, in a 10-D subspace, the complete time to perform normalization, projection, and search is around 1ms per image window on a DEC Alpha 3600. For a 512 × 480 image complete processing takes around 4 minutes. However, by applying rejection techniques such as [Baker and Nayar, 1996a] the overall time can be reduced to under 30secs.

5 Experimental Results

5.1 Feature Detection Rates

We statistically compare our step edge detector with the Canny [1986] and Nalwa-Binford [1986] detectors, following the approach in [Nalwa and Binford, 1986]. (See [Baker *et al.*, 1998] for more details.) Since we took great care modeling both the features and the imaging system, we used our step edge model to generate ideal step edges. For fairness, however, we changed the details slightly. Both the Canny and Nalwa-Binford detectors assume a constant blur/scale, so we fixed the value of σ in the step edge model to be 0.6 pixels. Secondly, the Nalwa-Binford detector is based on a square 5×5 window, as is Canny in the implementation that we used. Hence, we changed the window of our detector to be a square window containing 25 pixels, rather than the 49 pixel disc window used earlier. We generate "not edges" exactly as in [Nalwa and Binford, 1986], by taking a constant intensity window, and adding zero-mean Guassian noise.

In Figure 2 we compare the detection performance of the three edge detectors. For each pair of S.N.R. and detector, we plot a curve of false positives against false negatives obtained by varying the threshold inherent in each detection algorithm. The Canny operator thresholds on the gradient magnitude, the Nalwa-Binford detector thresholds on the estimated step size, and our approach thresholds on the distance from the parametric manifold. The rate of false positives was estimated by applying each detector to a constant intensity window with noise added. The rate of false negatives is obtained by applying the detectors to noisy ideal step edges.

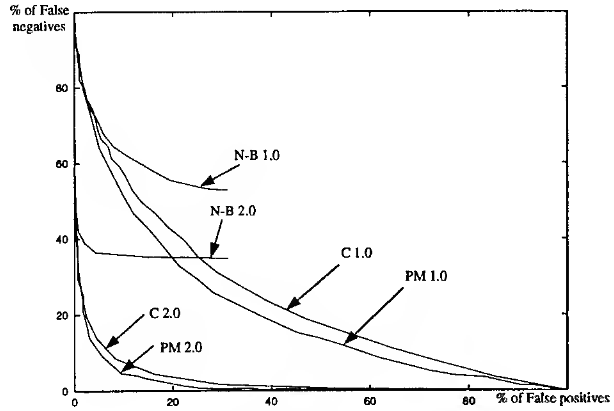


Figure 2: A comparison of edge detection rates. The Canny (C), Nalwa-Binford (N-B), and parametric manifold (PM) detectors are compared for S.N.R. = 1.0 and 2.0. We plot false positives against false negatives. For each detector and S.N.R., the result is a curve parameterized by the threshold inherent in that detector. The closer a curve lies to the origin, the better the performance. We see that the Canny detector and the parametric manifold technique perform comparably.

The closer a curve lies to the origin in Figure 2, the better the performance. Hence, we can see that both the Canny detector and our detector do increasingly well as the S.N.R. increases. The results for the Nalwa-Binford detector are consistent[3] with those described in [Nalwa and Binford, 1986]. Applied to real images, the Nalwa-Binford detector does not perform as poorly as Figure 2 might indicate. The poor Nalwa-Binford results are probably due to thresholding on the step-size and may well be completely different if we fix the step-size threshold, and vary the tanh-fit threshold.

5.2 Parameter Estimation Accuracy

Again following [Nalwa and Binford, 1986], we analyze parameter estimation accuracy by randomly generating a set of feature parameters, synthesizing a feature with these parameters, adding noise, applying the detector, and then measurings the accuracy of the estimated parameters. In Figure 3, we compare the performance of our step edge detector with that of the Canny detector [1986] and the Nalwa-Binford [1986] detector. In the figure, we plot the R.M.S. error in the estimate of the orientation θ against the S.N.R. We see that for low S.N.R. the performance of all detectors is limited by the noise. For lower noise levels, our detector marginally out-performs both of the other detectors.

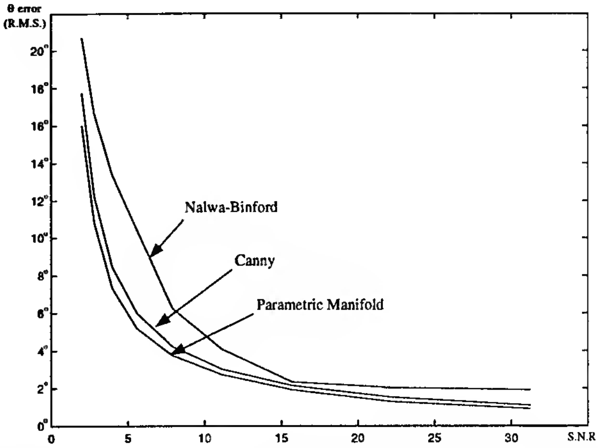


Figure 3: A comparison of the orientation estimation accuracy for the three step edge detectors. We took synthesized step edges, added noise to them, and then applied the edge detectors. We plot the R.M.S. error of the orientation estimate against the S.N.R. At all noise levels, the parametric manifold approach slightly outperforms both the Nalwa-Binford and Canny detectors.

5.3 Application to Images

In Figures 4(b) and (c) we present the results of applying our step edge and corner detectors to the image in Figure 4(a). The original image is taken from [MOMA, 1984] and was digitized using an Envisions 6600S scanner at 200dpi. We present the outputs of the detectors as grey-coded distance to the feature manifold (on a non-linear scale) so that the structure of the object can be seen clearly. It is immediate that the features detected are consistent with the original image. Thresholding on the distance to the feature manifold to finally detect features is straightforward as is demonstrated in [Baker *et al.*, 1998] where we superimpose thresholded feature maps on the original images.

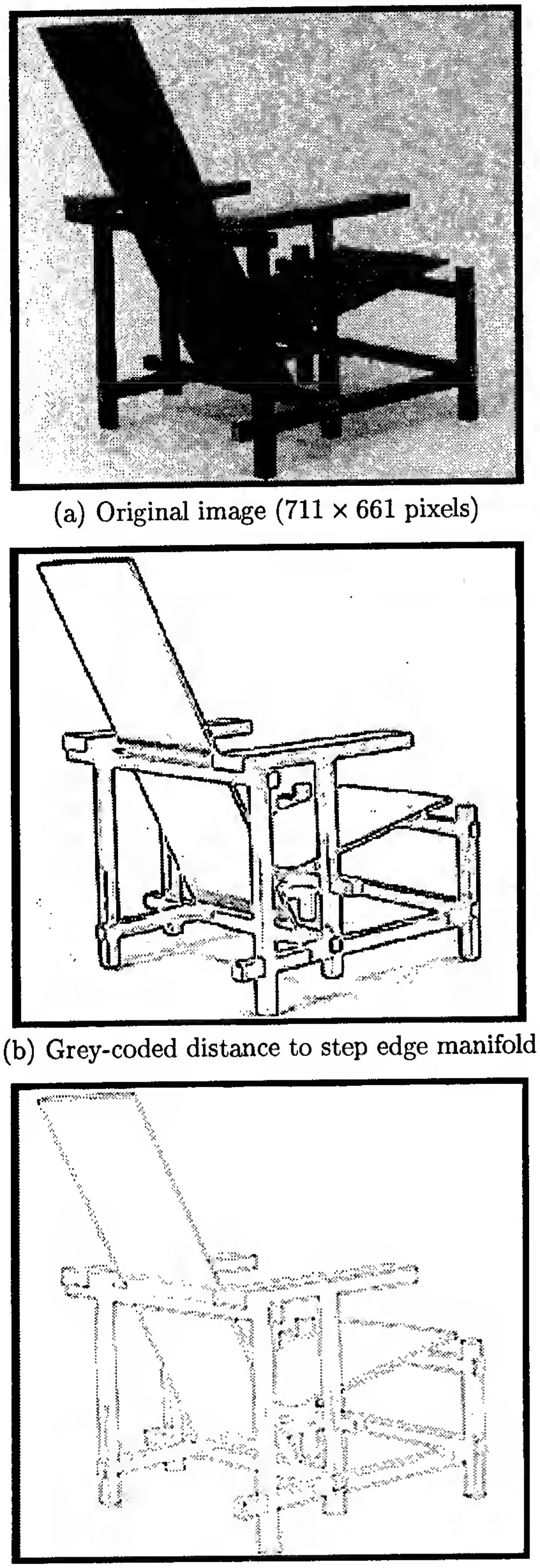
Acknowledgements

We wish to thank Vic Nalwa for providing an implementation of the Nalwa-Binford edge detector, and Sergio Cuniolo, Michael Heath, Tony Lindenberg, Jose-Maria Montiel, and Geoff West for providing pointers to Canny edge detectors.

References

[Baker and Nayar, 1996a] S. Baker and S.K. Nayar. Algorithms for pattern rejection. In *Proceedings of the IAPR Internation Conference*

[3] We did not use step 2)′ of the Nalwa-Binford algorithm, however the inclusion of this step does not radically alter the performance [Nalwa and Binford, 1986].



(a) Original image (711 × 661 pixels)

(b) Grey-coded distance to step edge manifold

(c) Grey-coded distance to corner manifold

Figure 4: Results of step edge and corner detection for a 711 × 661 image of "Red and Blue," by *Gerrit Rietveld*, circa 1918. The raw (unthresholded) detector outputs in (b) and (c) reflect high accuracy in detection and localization.

on Pattern Recognition, pages 869–874, 1996.

[Baker and Nayar, 1996b] S. Baker and S.K. Nayar. Pattern rejection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 544–549, 1996.

[Baker *et al.*, 1998] S. Baker, S.K. Nayar, and H. Murase. Parametric feature detection. *International Journal of Computer Vision*, 1998. (Accepted for publication).

[Barbe, 1980] D.F. Barbe. *Charge-Coupled Devices.* Spring-Verlag, 1980.

[Born and Wolf, 1965] M. Born and E. Wolf. *Principles of Optics.* Permagon Press, 1965.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[Fukunaga, 1990] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[Hueckel, 1971] M.H. Hueckel. An operator which locates edges in digitized pictures. *Journal of the Association for Computing Machinery*, 18:113–125, 1971.

[Hummel, 1979] R.A. Hummel. Feature detection using basis functions. *Computer Graphics and Image Processing*, 9:40–55, 1979.

[Koenderink, 1984] J.J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[Lenz, 1987] R. Lenz. Optimal filters for the detection of linear patterns in 2-D and higher dimensional images. *Pattern Recognition*, 20:163–172, 1987.

[MOMA, 1984] MOMA. *The Museum of Modern Art New York: The History and the Collection.* Harry N. Abrams, 1984.

[Nalwa and Binford, 1986] V.S. Nalwa and T.O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:699–714, 1986.

[Nandy *et al.*, 1996] D. Nandy, Z. Wang, J. Ben-Arie, K.R. Rao, and N. Jojic. A generalized feature extractor using expansion matching and the Karhunen-Loeve transform. In *Proceedings of the ARPA Image Understanding Workshop*, pages 969–972, 1996.

[Nayar *et al.*, 1996] S.K. Nayar, S. Baker, and H. Murase. Parametric feature detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 471–477, 1996.

[Norton, 1982] H.N. Norton. *Sensor and Analyzer Handbook.* Prentice-Hall, 1982.

Catadioptric Image Formation *

Shree K. Nayar and Simon Baker

Department of Computer Science, Columbia University
New York, New York 10027
Email: {nayar,simonb}@cs.columbia.edu

Abstract

Conventional video cameras have limited fields of view that make them restrictive in certain vision applications. A catadioptric sensor uses a combination of lenses and mirrors placed in a carefully designed configuration to capture a much wider field of view. In particular, the shape of the mirror must be selected to ensure that the complete catadioptric system has a single effective viewpoint, which is a requirement for the generation of pure perspective images from the sensed image. In this paper, we derive and analyze the complete class of single-lens single-mirror catadioptric sensors which satisfy the fixed viewpoint constraint. Some solutions turn out to be degenerate with no practical value while other solutions lead to realizable sensors.

1 Introduction

Conventional imaging systems are limited in their fields of view. An effective way to enhance the field of view is to use mirrors in conjunction with lenses. This approach to image formation is fast gaining in popularity (see [Nayar, 1988], [Yagi and Kawato, 1990], [Hong, 1991], [Goshtasby and Gruver, 1993], [Yamazawa *et al.*, 1993], [Nalwa, 1996] [Nayar, 1997]). We refer to the general approach of incorporating mirrors into conventional imaging systems as *catadioptric*[1] image formation. Our recent work in this context has led to the development of a truly omnidirectional video camera with a spherical field of view [Nayar, 1997].

As recently noted in [Yamazawa *et al.*, 1993], [Nalwa, 1996] and [Nayar, 1997], it is highly desirable that the catadioptric system (or any imaging system, for that matter) have a single center of projection (viewpoint). A single viewpoint permits the creation of pure perspective images from the image sensed by the catadioptric system. This is done by mapping sensed brightness values onto a plane placed at any distance (effective focal length) from the viewpoint. Any image computed in this manner preserves linear perspective geometry. For instance, straight lines in the scene produce straight lines in the computed image. Images that adhere to perspective projection are desirable from two standpoints; they are consistent with the way we are used to seeing images, and they lend themselves to further processing by the large body of work in computational vision that assumes linear perspective projection. When the catadioptric system is omnidirectional in its field of view, the single viewpoint permits the construction of not only perspective but also panoramic images.

In this paper, we derive the complete set of catadioptric systems with a single effective viewpoint and which are constructed from a single conventional lens and a single mirror. As we will show, the class of mirrors which can be used is exactly the class of rotated (swept) conic sections. Within this class of solutions, several swept conics prove to be degenerate solutions and hence impractical, while others lead to realizable sensors. During our analysis we will stop at many points to evaluate the merits of the solutions as well as the merits of catadioptric sensors proposed in the literature.

2 General Solution

Let the final (dioptric) stage of our sensor be a conventional perspective lens. In Figure 1, the effective pinhole of the lens is **p**. We formulate the catadioptic image formation problem as follows:

*This work was supported in parts by the DARPA/ONR MURI Grant N00014-95-1-0601, an NSF National Young Investigator Award, and a David and Lucile Packard Fellowship.

[1] *Dioptrics* is the optics of refracting elements (say, lenses) whereas *catoptrics* is the optics of reflecting surfaces (mirrors). The combination of refracting and reflecting elements is referred to as *catadioptrics* [Hecht and Zajac, 1974].

Find the class of reflecting surfaces that, when used in conjunction with a perspective lens, produce an image of the world as seen from a fixed viewpoint. Let us assume that the fixed viewpoint **v** is at the origin of the coordinate frame (see Figure 1) and the center **p** of the perspective lens is located on the vertical axis at a distance c from **v**.

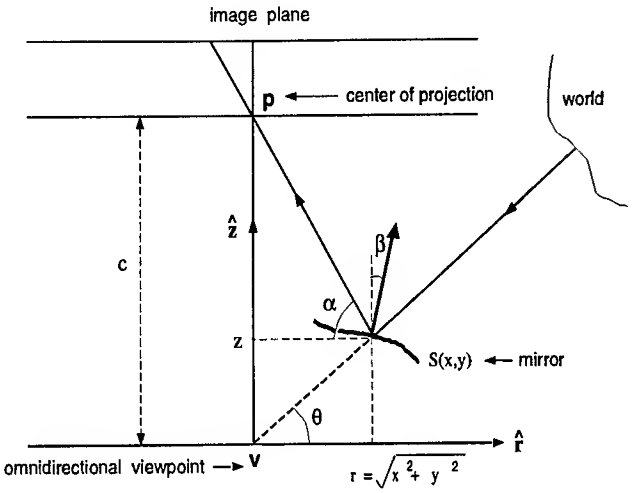


Figure 1: Geometry used to derive the reflecting surface that produces an image of the world as seen from a fixed viewpoint **v**. This image is captured using a conventional perspective camera with an effective pinhole **p**.

For the fixed viewpoint constraint to hold, each world point seen from **v** must be reflected by a point on the mirror surface $S(x, y)$ towards **p**. Note that since perspective projection is rotationally symmetric about the optical axis $\hat{\mathbf{z}}$, the mirror can be assumed to be a surface of revolution around $\hat{\mathbf{z}}$. Therefore, it suffices to find the one-dimensional profile $z(r) = S(x, y)$, where $r = \sqrt{x^2 + y^2}$.

In fact, the viewpoint **v** and the determined profile $z(r)$ are in no way restricted to the optical axis. Since perspective projection is rotationally symmetric with respect to any ray that passes through the pinhole **p**, the viewpoint and the profile could be moved from the optical axis by keeping the distance c the same and aligning the symmetry axis of the profile with the ray that passes through the viewpoint and the pinhole. Of course, in this case, the image plane shown in Figure 1 would be non-frontal. This does not pose any additional ambiguity as the mapping from any non-frontal image plane to a frontal image plane is one-to-one.

With the above generalizations in place, we are ready to derive the profile of the reflecting surface. The relation between the angle θ of the incoming ray and the reflecting surface is

$$\tan\theta \;=\; \frac{z}{r}\,. \qquad (1)$$

The angle α made by the reflected ray with the horizontal axis is given by

$$\tan\alpha \;=\; \frac{(c - z)}{r}\,. \qquad (2)$$

Let the surface slope at the point of reflection be defined by the angle β made by the surface normal with the vertical axis:

$$\frac{dz}{dr} \;=\; -\tan\beta\,. \qquad (3)$$

This allows us to write

$$\tan 2\beta \;=\; \frac{2\tan\beta}{1 - \tan^2\beta} \;=\; \frac{-2\frac{dz}{dr}}{1 - (\frac{dz}{dr})^2}\,. \qquad (4)$$

Since the surface is *specular*, the angles of incidence and reflection are equal. Consequently,

$$\beta \;=\; \frac{(\alpha - \theta)}{2}\,, \qquad (5)$$

which gives us

$$\tan 2\beta \;=\; \frac{\tan\alpha \,-\, \tan\theta}{1 \,+\, \tan\alpha\,\tan\theta}\,. \qquad (6)$$

In the right hand side of the above expression, we substitute (1) and (2) and equate with the right hand side of (4) to get

$$\frac{-2\frac{dz}{dr}}{1 - (\frac{dz}{dr})^2} \;=\; \frac{(c \,-\, 2z)\,r}{(r^2 \,+\, c\,z \,-\, z^2)}\,. \qquad (7)$$

Thus, we find that the reflecting surface must satisfy the above quadratic first-order differential equation. It is straightforward to solve the quadratic for surface slope:

$$\frac{dz}{dr} = \frac{(z^2 - r^2 - c\,z) \pm \sqrt{r^2c^2 + (z^2 + r^2 - cz)^2}}{r\,(2z - c)} \qquad (8)$$

Next, we substitute $y = z - c/2$ and set $b = c/2$ which yields

$$\frac{dy}{dr} = \frac{(y^2 - r^2 - b^2) \pm \sqrt{4r^2b^2 + (y^2 + r^2 - b^2)^2}}{2ry}. \qquad (9)$$

Then, substituting $2rx = y^2 + r^2 - b^2$, we get

$$\frac{1}{\sqrt{b^2+x^2}}\frac{dx}{dr} \;=\; \pm\frac{1}{r}. \qquad (10)$$

Integrating both sides results in

$$\ln\left(x+\sqrt{b^2+x^2}\right) \;=\; \pm\ln r + C \qquad (11)$$

where, C is the constant of integration. Hence,

$$x+\sqrt{b^2+x^2} \;=\; \frac{k}{2}r^{\pm 1} \qquad (12)$$

where, $k = 2e^C > 0$ is a constant. By back substituting and simplifying we arrive at two equations which comprise the general solution:

$$\left(z-\frac{c}{2}\right)^2 + r^2\left(1-\frac{k}{2}\right) \;=\; \frac{c^2}{4}\left(\frac{k-2}{k}\right), \quad (13)$$

$$\left(z-\frac{c}{2}\right)^2 + r^2\left(1+\frac{c^2}{2k}\right) \;=\; \left(\frac{2k+c^2}{4}\right). \quad (14)$$

Together, these two expressions represent the entire class of mirrors that satisfy the fixed viewpoint constraint. Again, since perspective projection is symmetric about any ray that passes through the pinhole, the viewpoint **v** and the corresponding mirror are in no way restricted to the optical axis.

3 Specific Mirrors

A quick glance at the forms of equations (13) and (14) reveals that the mirror profiles are conic sections. However, each conic section must be placed at a specific distance from the pinhole. As we shall see, though all our conic sections are theoretically valid, many prove to be impractical and only a few lead to useful solutions. We are now in a position to evaluate several specific cases.

3.1 Planes

In solution (13), if we set $k = 2$, we get the cross-section of a planar mirror:

$$z \;=\; \frac{c}{2}\,. \qquad (15)$$

As shown in Figure 2, the plane bisects the line segment joining the pinhole and the viewpoint. This result is easily generalized to arbitrary planes or viewpoints. For any plane with unit normal $\hat{\mathbf{n}}$ and any point $\mathbf{q}$ on it, the viewpoint is simply the reflection of the pinhole

$$\mathbf{v} \;=\; \mathbf{p} \;-\; 2\left(\left(\mathbf{p}-\mathbf{q}\right)\cdot\hat{\mathbf{n}}\right)\hat{\mathbf{n}}\,. \qquad (16)$$

Equivalently, for any desired viewpoint, points $\mathbf{x}$ on the planar mirror are given by

$$\left(\mathbf{x}-\frac{(\mathbf{p}+\mathbf{v})}{2}\right)\cdot(\mathbf{p}-\mathbf{v}) \;=\; 0\,. \qquad (17)$$

These expressions lead us to a simple but unfortunate theorem: For a single fixed pinhole, no two planar mirrors can share the same viewpoint, and equivalently, two different viewpoints cannot be generated by the same planar mirror. It is clear from Figure 2 that a single planar mirror does not enhance the field of view of the imaging system. At the same time, the above theorem makes it impossible to increase the field of view by packing a large number of planar mirrors (pointing in different directions) in front of a conventional imaging system. On the brighter side, the two views of a scene needed for stereo can be captured by a single lens and two planar mirrors, as shown in [Goshtasby and Gruver, 1993].

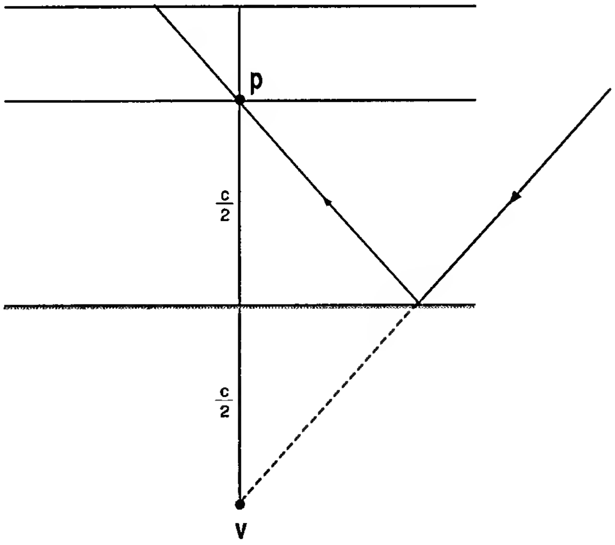


Figure 2: A planar mirror must bisect the segment joining the pinhole and the desired viewpoint. Since two planar mirrors cannot generate the same viewpoint, multiple planar mirrors cannot be used to enhance the field of view of a conventional imaging system.

To ensure a single viewpoint while using multiple planar mirrors, Nalwa [Nalwa, 1996] has arrived at a clever design that includes four planar mirrors that form the faces of a pyramid. Four separate imaging systems are used, each one placed above one of the faces of the pyramid. The optical axes of the imaging systems and the angles made by the four planar faces are adjusted so that the four viewpoints produced by the planar mirrors coincide. The result is a sensor that has a single viewpoint and a panoramic field of view of approximately 360°×50°. The panoramic image is of relatively high resolution as it is a concatenation of four images provided by four non-overlapping imaging systems. Nalwa's sensor is easy enough to implement but requires the use of four of each component (cameras, lenses, and digitizers).

3.2 Cones

In solution (13), if we set $c = 0$, the result is a conical mirror with cross-section

$$z = \sqrt{\frac{k-2}{2} r^2}, \qquad k \geq 2 . \qquad (18)$$

The angle at the apex of the cone varies with k. At first glance, this may seem like a reasonable solution. However, since $c = 0$, the apex of the cone must be at the pinhole. This implies that the rays of light entering the pinhole can only graze the cone and do not represent reflections of the world (see Figure 3). Hence, we have a degenerate solution that is of no practical value.

Indeed, the cone has been used for wide-angle imaging, in particular, for autonomous navigation [Yagi and Kawato, 1990]. In these implementations, the apex of the cone was placed at a distance from the pinhole. In such cases, it is easy to show that the viewpoint is no longer a single point but rather a locus [Nalwa, 1996]. If the axis of the cone points in the direction of the pinhole, the locus is a circle that hangs like a halo around the cone.

3.3 Spheres

In solution (14), if we set $c = 0$, we get a spherical mirror with cross-section

$$z^2 + r^2 = \frac{k}{2}, \qquad k > 0 . \qquad (19)$$

Like the cone, this proves to be a solution of little value; since the viewpoint and pinhole must coincide, the observer sees itself and nothing else, as shown in Figure 4.

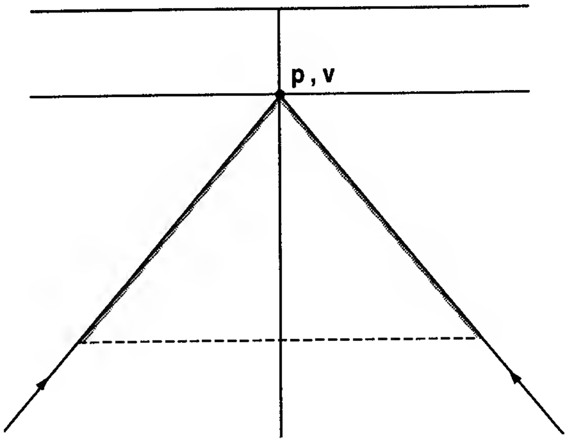


Figure 3: The conical mirror has its apex at the pinhole. This degenerate solution is of little practical value. If the apex is moved away from the pinhole, the viewpoint is no longer single but rather lies on a circular locus.

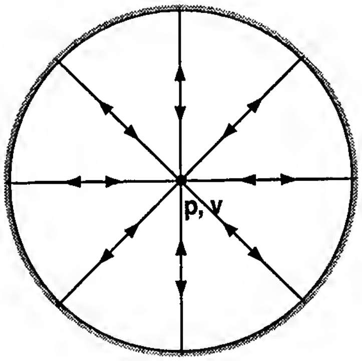


Figure 4: A spherical mirror produces a single viewpoint only when the pinhole lies at its center. This, again, is a solution of little use as the observer sees itself and nothing else.

In [Hong, 1991], a wide-angle implementation with a sphere is described that was used for landmark navigation. In this case, the sphere was placed at a distance from the effective pinhole of the camera. As with the cone, the result is a locus of viewpoints rather than a single viewpoint. The locus in the case of the sphere can turn out to be a surface of large extent, depending on the distance between the center of the sphere and the pinhole. In [Nayar, 1988], a stereo system is proposed that uses a single image of two specular spheres to compute depth. In this case, the single viewpoint constraint is not critical as stereo requires multiple viewpoints.

3.4 Ellipsoids

In solution (14), when $c > 0$ and $k > 0$, we get an ellipsoid with cross-section

$$\frac{(z-\frac{c}{2})^2}{a^2} + \frac{r^2}{b^2} = 1 \, , \qquad (20)$$

where

$$a = \sqrt{\frac{2k+c^2}{4}} \, , \quad b = \sqrt{\frac{2k}{4}} \, . \qquad (21)$$

We have now arrived at a solution that can be used to enhance the field of view. As shown in Figure 5, the viewpoint **v** and pinhole **p** are located at the two foci of the ellipse, respectively. If, for instance, the section of the ellipse that lies beneath the viewpoint is used, the effective field of view (ignoring self-occlusion by the lens) corresponds to the upper hemisphere. It is easy to see that terminating the ellipse below the viewpoint does not enhance the field of view. Unfortunately, extending the mirror above the viewpoint does to help either; in this case, rays of light entering the pinhole would have undergone multiple reflections by the mirror. Yet, the ellipse does represent our first useful solution. It is similar in nature to our next solution, the hyperboloid. Hence, we shall defer our discussion on implementation issues related to the ellipsoid.

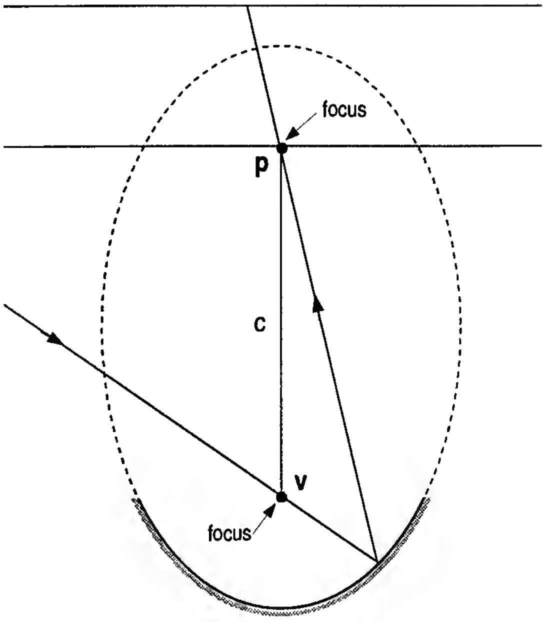


Figure 5: A ellipsoidal mirror is a viable solution when the pinhole and the viewpoint are located at its two foci, respectively. If the ellipsoid is terminated by a plane passing through the viewpoint, the field of view corresponds a hemisphere.

3.5 Hyperboloids

In solution (13), when $c > 0$ and $k > 2$, we get a hyperboloid[2] with cross-section

$$\frac{(z-\frac{c}{2})^2}{a^2} - \frac{r^2}{b^2} = 1 \, , \qquad (22)$$

where

$$a = \frac{c}{2}\sqrt{\frac{k-2}{k}} \, , \quad b = \frac{c}{2}\sqrt{\frac{2}{k}} \, . \qquad (23)$$

As we see in Figure 6, in the limit $k \rightarrow 2$, the hyperboloid flattens to yield the planar solution of section 3.1. As k increases, the curvature of the hyperboloid increases, and hence also the field of view of the catadioptric system. The two foci of

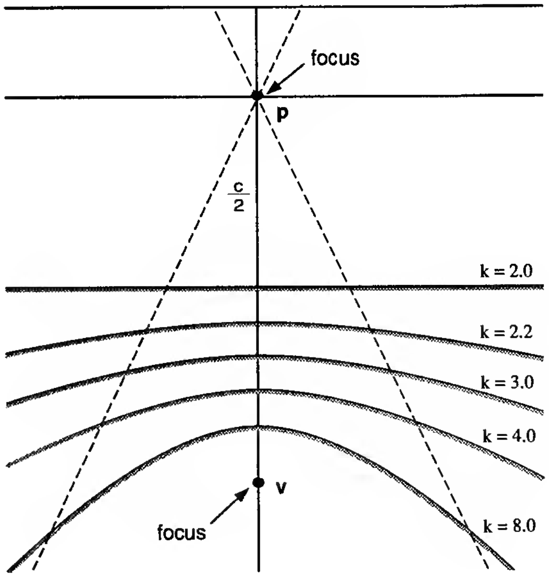


Figure 6: The hyperboloidal mirror can produce the desired increase in field of view. The pinhole and the viewpoint are located at the two hyperboloidal foci.

the hyperboloid remain fixed, one at the pinhole $\mathbf{p}$ and the other at the viewpoint $\mathbf{v}$.

This solution provides a practical approach to wide-angle imaging. Yamazawa et al. [Yamazawa *et al.*, 1993] recognized that the hyperboloid, if chosen and positioned carefully, would produce a single viewpoint. They implemented a sensor for autonomous navigation and demonstrated the construction of perspective images from hyperboloidal ones.

While this solution is both interesting and feasible, it must be implemented with care. As can be seen from Figure 6, for any chosen value of k, if the viewpoint is distant from the pinhole, the mirror must be large. As the viewpoint approaches the pinhole, the mirror reduces in size but the curvatures at all points on the mirror increase. This increases the optical effects of coma and astigmatism that are known to produce blurring [Hecht and Zajac, 1974]. Furthermore, it is hard to configure an imaging system with a large enough depth of field close to the pinhole that would allow the hyperboloidal mirror to be placed in close proximity. These trade-offs imply that the distance of the viewpoint must be chosen with care. Also, the axis of the hyperboloid must pass through the pinhole. For these reasons, careful implementation is required to achieve the desired optical properties and precise calibration is needed to establish the mapping between an incoming principle ray and its image coordinates. It is easy to see that all of the above implementational issues also apply to the ellipsoidal solution.

3.6 Paraboloids

If image projection is orthographic rather than perspective, the geometrical mappings between the image, the mirror and the world are invariant to translations of the mirror with respect to the imaging system. Consequently, both calibration as well as the computation of perspective images are greatly simplified. There are simple ways to achieve pure orthographic projection, as described in [Nayar, 1997].

The shape of the mirror in this case can be derived by assuming orthographic projection rather than perspective projection in the dioptric stage of image formation. This derivation is given in [Nayar, 1997] and the mirror is shown to be paraboloidal (see Figure 7). Paraboloidal mirrors are frequently used to converge an incoming set of parallel rays at a single point (the focus), or to generate a collimated light source from a point source (placed at the focus). In both these cases, the paraboloid is a concave mirror that is reflective on its inner surface. In our case, the paraboloid is reflective on its outer surface (convex mirror); all incoming principle rays are orthographically reflected by the mirror. Further, the incoming rays can be extended to intersect at the focus of the paraboloid, which serves as the viewpoint.

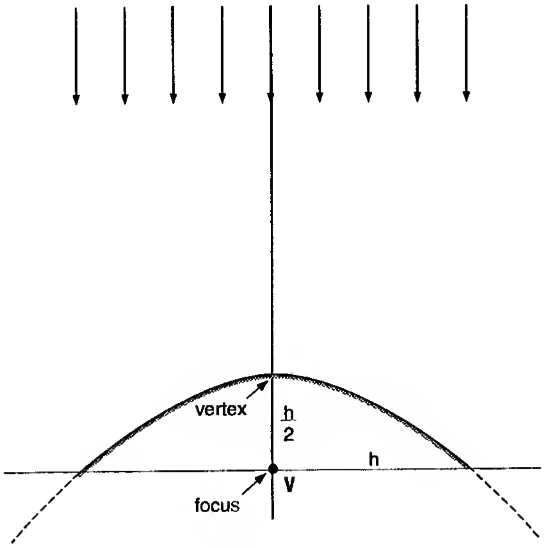


Figure 7: For orthographic projection, the solution is a paraboloid with the viewpoint located at the focus. Orthographic projection makes the geometric mappings between the image, the paraboloidal mirror and the world invariant to translations of the mirror. This greatly simplifies calibration and the computation of perspective images from paraboloidal ones.

Alternatively, the same solution can be derived from our general solution for catadioptric imaging. We know that orthographic projection is a limiting case of perspective, where the distance between the pinhole and viewpoint approaches infinity. Equation (13) can be rewritten as:

$$\frac{z^2}{k} - \frac{zc}{k} + \frac{r^2}{k} - \frac{1}{2}r^2 \;=\; -\frac{c^2}{2k^2}. \qquad (24)$$

Then, in the limit $c \to \infty$, $k \to \infty$, while keeping $c/k = h$ a constant, we have

$$z \;=\; \frac{h^2 - r^2}{2h}. \qquad (25)$$

[2] Note that $k < 2$ is not possible in solution (13) since this yields an imaginary surface.

The parameter h of the paraboloid is its radius at $z = 0$. The distance between the vertex and the focus is $h/2$. Therefore, h determines the size of the paraboloid that, for any given orthographic lens system, can be chosen to maximize resolution. If the paraboloid is terminated at its focus, the imaging system yields a hemispherical field of view. As shown in [Nayar, 1997], two such imaging systems can be placed back-to-back to achieve a spherical field of view.

4 Resolution

Here, we define resolution as the solid angle subtended from the viewpoint by a pixel in the sensed image. Let us assume that the area projected by a pixel along its line of sight is da, as shown in Figure 8. Note that for orthographic projection da is a constant, while for perspective projection it is easily computed from the distance of the corresponding point on the mirror and the focal length of the imaging system. Given that all reflections are specular, the reflecting surface area occupied by da is $ds = da/\cos\phi$. The foreshortened surface area as seen by the viewpoint $\mathbf{v}$ is $(da/\cos\phi)\cos\phi = da$. The solid angle subtended by the reflecting surface element is $d\omega = da/t^2$, where t is the distance of the surface element from the viewpoint. Hence, the spatial resolution for any catadioptric sensor can be written as

$$\frac{da}{d\omega} = t^2 = \sqrt{z^2 + r^2}\,. \qquad (26)$$

For instance, in the case of a paraboloidal mirror [Nayar, 1997], the resolution increases by a factor of 4 from the vertex ($t = h/2$) of the paraboloid to the fringe ($t = h$). With (26), it is easy to see that a variation in spatial resolution occurs not only in the case of curved mirrors but also planar ones. In principle, it is of course possible to use image detectors with non-uniform resolution to compensate for the above variation.

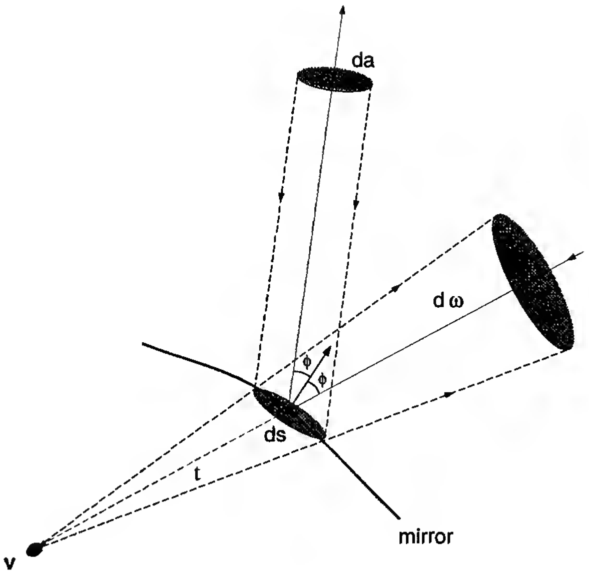


Figure 8: Geometry used to derive spatial resolution ($da/d\omega$) for any catadioptric sensor.

References

[Goshtasby and Gruver, 1993] A. Goshtasby and W. A. Gruver. Design of a Single-Lens Stereo Camera System. *Pattern Recognition*, 26(6):923–937, 1993.

[Hecht and Zajac, 1974] E. Hecht and A. Zajac. *Optics.* Addison Wesley, Reading, Massachusetts, 1974.

[Hong, 1991] J. Hong. Image Based Homing. *Proc. of IEEE International Conference on Robotics and Automation*, May 1991.

[Nalwa, 1996] V. Nalwa. A True Omnidirectional Viewer. Technical report, Bell Laboratories, Holmdel, NJ 07733, U.S.A., February 1996.

[Nayar, 1988] S. K. Nayar. Sphereo: Recovering depth using a single camera and two specular spheres. *Proc. of SPIE: Optics, Illumumination, and Image Sensing for Machine Vision II*, November 1988.

[Nayar, 1997] S. K. Nayar. Omnidirectional Video Camera. *Proc. of DARPA Image Understanding Workshop*, May 1997.

[Yagi and Kawato, 1990] Y. Yagi and S. Kawato. Panoramic Scene Analysis with Conic Projection. *Proc. of International Conference on Robots and Systems (IROS)*, 1990.

[Yamazawa *et al.*, 1993] K. Yamazawa, Y. Yagi, and M. Yachida. Omnidirectional Imaging with Hyperboloidal Projection. *Proc. of International Conference on Robots and Systems (IROS)*, 1993.

Imaging-Consistent Super-Resolution*

Ming-Chao Chiang

Columbia University
Department of Computer Science
New York, NY 10027
chiang@cs.columbia.edu

Terrance E. Boult

Lehigh University
Department of EECS
Bethlehem, PA 18015
tboult@eecs.lehigh.edu

Abstract

This paper introduces two algorithms for enhancing image resolution from an image sequence. The "image-based" approach presumes that the images were taken under the same illumination conditions and uses the intensity information provided by the image sequence to construct the high-resolution image. When imaging from different viewpoints, over long temporal spans, or imaging scenes with moving 3D objects, the image intensities naturally vary. The "edge-based" approach, based on edge/blur models, allows super-resolution under lighting variations. The paper presents the theory and the experimental results using these two algorithms.

1 Introduction

The idea of super-resolution, combining images by combining pieces from an image sequence into a single image with higher resolution than any of the individual images, has been around for years. Previous research on super-resolution, [Huang and Tsai-1984, Gross-1986, Peleg *et al.*-1987, Keren *et al.*-1988, Irani and Peleg-1991, Irani and Peleg-1993, Bascle *et al.*-1996], ignores the impact of image warping techniques. It also presumes that the images were taken under the same illumination conditions. This paper summarizes our recent work which addresses techniques to improve the quality of super-resolution imaging and to deal with lighting variations. We show that image warping techniques may have a strong impact on the quality of image resolution enhancement.

Image warping requires the underlying image to be "resampled" at non-integer locations; it requires spatially varying image reconstruction. When the goal of warping is to produce output for human viewing, only mildly accurate image intensities are needed. In these cases, techniques using bi-linear interpolation have been found sufficient. However, as a step for applications such as super-resolution, the precision of the warped intensity values is often important. For these problems, bi-linear image reconstruction may not be sufficient; the spatially varying nature of the reconstruction limits the "efficient" alternative reconstruction methods. This paper shows how ideas of imaging-consistent reconstruction/restoration algorithms [Boult and Wolberg-1993] and the intcgrating resampler [Chiang and Boult-1996b] can be used for warping while maintaining superior image quality.

2 Image-Based Super-Resolution

The idea of super-resolution is based on the fact that each image in the sequence provides small amount of additional information. There are, of course, some fundamental limits on what this combination can do. If the images were noise-free, focused and Nyquist sampled, then multiple images would add nothing. However, if the images are blurred and with the noise, and aliasing presents in images, deblurring is unstable. If time is not a concern, then standard DSP techniques can address these problems, formulating fusion as millions of coupled equations. The goal is then to come up with an efficient approximation.

Our approach recognizes four separate components: the matching (to determine alignment), the warping (to align the data and increase sampling rate), the fusion (to produce a less noisy image), and an optional deblurring stage to remove lens blur. For now, we are using traditional matching on image fields (normalized SSD or correlation) and traditional deblurring. We are concentrating our efforts on warping and fu-

*This work is supported in part by NSF PYI IRI-90-57951, NSF grant CDA-9413782, and ONR/DARPA via DOD MURI program ONR N00014-95-1-0601. Several other agencies and companies have also supported parts of this research.

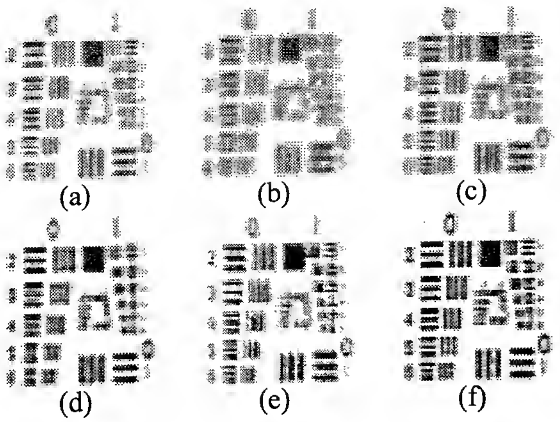


Figure 1: Original images and down-sampled version of super-resolution results. (a) shows one of the eight original images. (b) shows the down-sampled super-resolution using bi-linear resampling. (c) down-sampled super-resolution using QRS (d) shows a deblurred original (i.e., deblurred (a). (e) shows down-sampled super-resolution by back-projection (f) shows super-resolution with QRS followed by deblurring followed by down-sampling.

sion. Warping is considered in the next section.

For fusion, we have experimented with simple averaging, averaging with trimmed tails, and median. These produce decreasingly accurate estimates with increasing robustness to outliers. As the matching is sometimes inaccurate and because of aliasing artifacts, a few outliers are common, thus the trimmed tails is probably the best overall technique.

In [Chiang and Boult-1996a], we presented initial results and compared our technique to the leading existing work of [Irani and Peleg-1993] (which is referred as back-projection in the following). Figs. 1, 2, and 3 show some example results. In all cases, the resulting super-resolution images are a scale-up by a factor of 4. We note that previous work on this topic reported results only scaling by a factor of 2.

If we down-sample our the super-resolution estimation, we should get an increase in image quality. A few examples of this are shown in Fig. 1. It can be easily seen from Fig. 1 that image warping techniques indeed have a strong impact on the quality enhancement, even if the image resolution is not increased. In particular, Fig. 1f is significantly clearer than the original (Fig. 1a) or a deblurred version thereof (Fig. 1d). Thus, super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.

Fig. 2 shows the final results of our first experiment. Fig. 2a shows Fig. 1a blown up by a factor

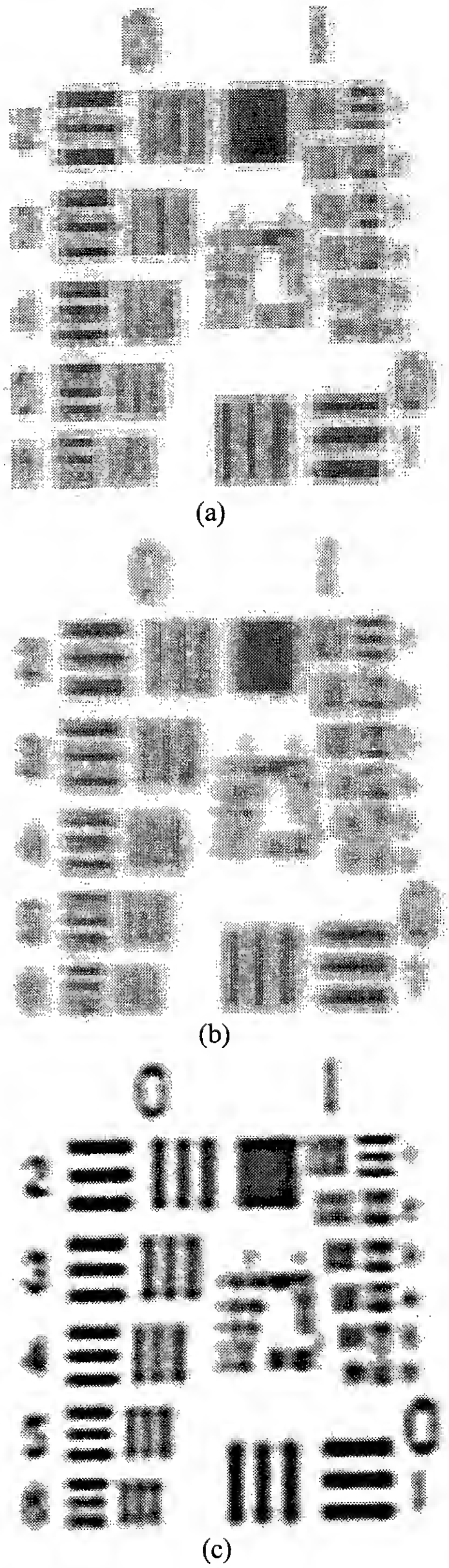
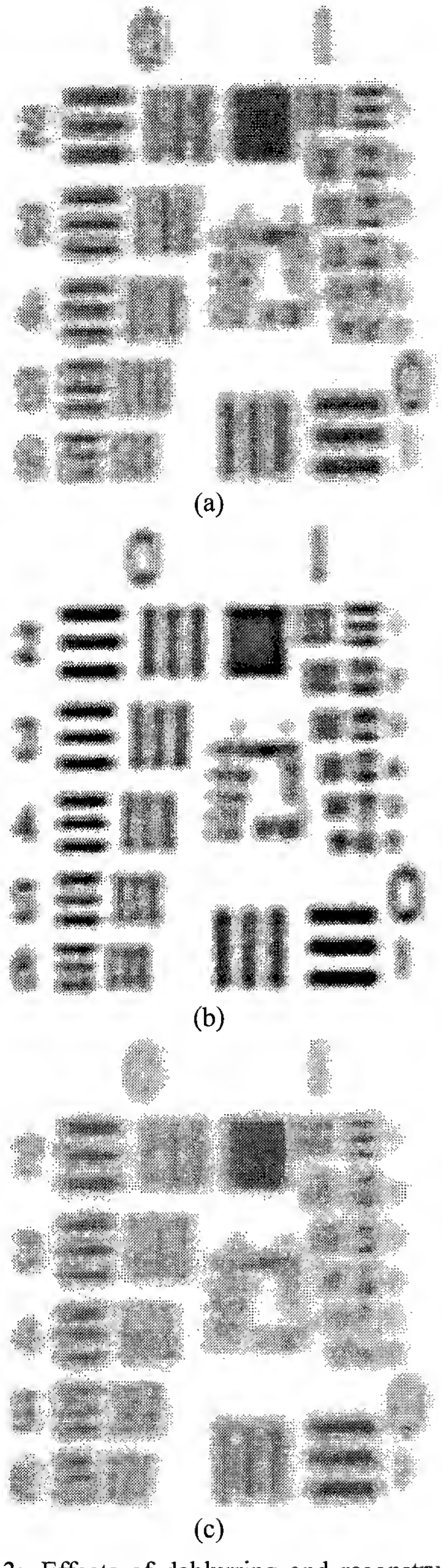


Figure 2: Fig. 1a pixel-replicated by a factor of 4; (b) super-resolution by back-projection; (c) super-resolution using QRS with deblurring.



(a)

(b)

(c)

Figure 3: Effects of deblurring and reconstruction kernel. (a) shows Fig. 2(c) without deblurring. (b) shows super-resolution using bi-linear resampling with deblurring; and (c) shows (b) before deblurring.

of 4 using pixel replication. Fig. 2b shows super-resolution by our implementation of Irani's back-projection method using bi-linear resampling to simulate the image formation process. Fig. 2c shows super-resolution using QRS followed by deblurring. Fig. 3 shows the effects of deblurring and of using bi-linear reconstruction for warping.

Fig. 4 shows our second example—a more complex gray level image. The tread-wheels of the toy tank are visible in the super-resolution image but not in the originals, and the "tank-number" is (just) readable in the super-resolution image while not in the original.

Results from our experiments show that the image-based method we propose herein is not only computationally cheaper, but it also gives results comparable to or better than those using back-projection. In general, our method is often more than two or three times faster. See [Chiang and Boult-1996a, Chiang-1996] for details. Moreover, it is easily seen from Figs. 1, 2, and 3 that integrating resampler outperforms traditional bi-linear resampling.

3 Imaging-Consistent Warping

Central in our imaging-based technique is the use of what we call imaging-consistent warping. Due to the limit of space, we only briefly review the this idea; more details (including the image formation process and the sensor model) can be found in [Chiang and Boult-1996b].

Consider the imaging model in Fig. 5. An algorithm is called *imaging-consistent* if it is the exact solution for some input function, which, according to the sensor model, would have generated the measured input. For image reconstruction, we achieve this by computing a functional restoration (i.e., f_2), then blurring it again by the pixel's PSF. This actually defines a whole class of image restoration/reconstruction techniques, depending on the model for f_2. Probably the simplest method to consider is based on a piecewise quadratic model for the image. If we assume a Rect PSF filter for the photosite, the imaging consistent algorithm is easy to derive (see [Chiang and Boult-1996b]). To ensure that the function is continuous, and that the method is local, we define the value of the reconstruction at the pixel boundaries k_i and k_{i+1}, to be equal to E_i and E_{i+1}, where we compute E_i with some techniques, e.g., cubic convolution. The values E_i at the pixel edges, combined with the imaging-consistent constraint (the integral across the pixel must equal the measured intensity) result in exactly three constraints. From this, one can determine

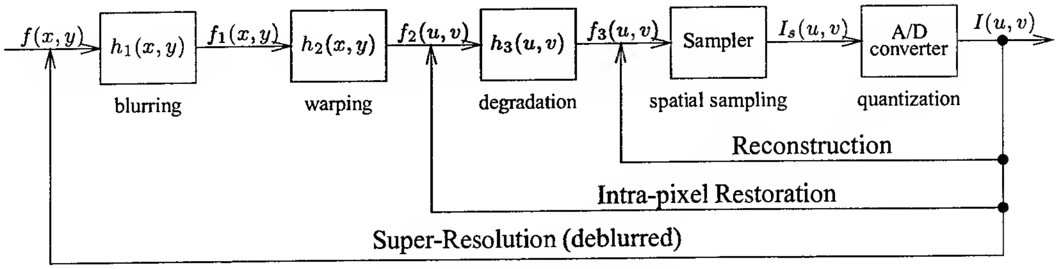


Figure 5: The image formation process and the relationship between restoration, reconstruction, and super-resolution.

the quadratic polynomial for f_2. This gives the intra-pixel restoration. For super-resolution, we consider only this intra-pixel restoration (abbreviated QRS in the following discussion). Reconstruction can be derived by simply blurring the resulting restoration by a PSF of the same scale as input.

To define the integrating resamplers, we generalize the idea of the imaging-consistent algorithms described above. Whereas imaging-consistent algorithms simply assume the degradation models are identical for both input and output; the integrating resamplers go one step further, allowing (1) both input and output to have their own degradation model, and (2) the degradation model to vary its size for each output pixel.

When we are resampling the image and warping its geometry in a nonlinear manner, this new approach allows us to efficiently do both pre-filtering and post-filtering. Because we have already determined a functional form for the input, no spatially-varying filtering is needed, as would be the case if direct inverse mapping were done. The integrating resampler [Chiang and Boult-1996b] also handles antialiasing of partial pixels in a straightforward manner.

4 Edge-Based Super-Resolution

For almost all applications involving an image sequence, the problem of lighting variation arises, even when they are taken consecutively in a well controlled environment. If the images are not from a short time span, variations are often significant. The idea we propose herein is a simple solution, we fuse edge/blur information with one of the original intensity image to reconstruct the super-resolution image. This effectively mitigates the problem of lighting variation since edge positions are much less sensitive to a change of lighting.

To fuse all the edges together, it is required that the edges first be detected and warped. It also requires an image reconstruction technique that directly incorporates both the edge and intensities. This will allow the reference image to be reestimated and scaled up based on the edge models and local blur estimation. We have generalized the idea of the imaging-consistent reconstruction algorithms to deal with discontinuities in an image [Chiang and Boult-1997].

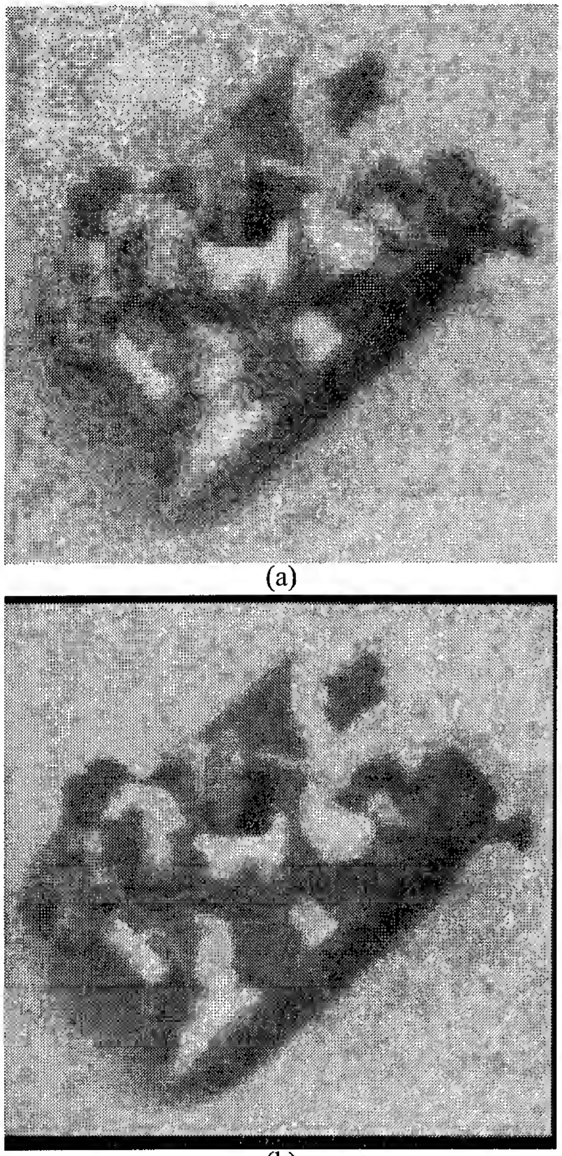
Given the image sequence, our edge-based super-resolution algorithm is shown, as follows:

1. Estimate the edges and blur models using the procedure described in Section 4.1.
2. Estimate the motions involved in the image sequence.
3. Choose one of the images as the reference image (the one with "best" lighting). Scale the reference image up and deblur at the same time it is being scaled up.
4. Warp all the edges/blur models to the reference image and fuse them.
5. Use the fused edge/blur models and the reference image to compute the super-resolution intensity image.
6. Optional deblurring stage.

4.1 Edge Localization& Local Blur Estimation

Typically, edge detection involves the estimation of first and second derivatives of the luminance function, followed by selection of zero crossing and extrema. While the world does not need yet another edge detector, we define a new one because it allows us to work in a consistent framework, incorporating the edge model into the image reconstruction algorithm. The edge localization/detection is obtained by differentiating the functional form of the image reconstruction model and considering only significant extrema of the first derivative.[1] The edge model is

[1] Yea, there is a threshold hiding there. Future work will address how to better determine significant vs. insignificant edges, and, more importantly, if this should be done before or after the fusion of the "edge-models".



(a)

(b)

Figure 4: Results from a toy-tank sequence. (a) one of the original images pixel replicated by a factor of 4; (b) super-resolution using QRS followed by deblurring.

tied into the image reconstruction in that each pixel is now modeled as potentially having a discontinuity, while still satisfying the imaging-consistent constraint. The model we use is piecewise quadratic, and the integral across the pixel, including any step discontinuity, must still equal the measured data. If a discontinuity is included within a pixel, the approximations used for the pixel boundaries are recomputed using data from only one side of the discontinuity.

In [Chiang and Boult-1996a], we showed that deblurring after image fusion is most effective for super-resolution imaging. However, that work presumes that the blur is not dominated by depth-of-field effects. This allows us to replace a spatially-varying point spread function with a cascade of two simpler components: a spatially-invariant blur and a geometric warp. Unfortunately, this assumption is rarely true in practice.

In [Chiang and Boult-1997] we describe our local blur estimation in detail. In summary, we use a simple blurred step edge, assuming that the step edge is from v to $v + \delta u(x)$ where v is the unknown intensity value and δ is the unknown amplitude of the edge. The blur of this edge is modeled by a "truncated" Gaussian blur kernel

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

where σ is the unknown standard deviation. The complete edge model is thus given by Eq. (1) and Eq. (2) where α is a predefined nonnegative constant, and $x_0 \in [0, 1]$ is the location of the step edge. Given the functional form of our reconstruction, we can solve directly for the three parameters of the blurred edge model.

For the examples, we again presume that "motion" is computed, which for general lighting changes is much more difficult. For the examples, here we use a normalized SSD computation. Fig. 6 shows two of the eight original images with two different illumination conditions blown up by a factor of 4 using, respectively, pixel replication and bi-linear resampling. Figs. 6a and b shows originals magnified with pixel replication, while Figs. 6c and d, show the edge-based super-resolution results before and after deblurring.

5 Image-Based vs. Edge-Based

Due to the limit of space, we only briefly compare the two algorithms proposed herein. Both algorithms take time roughly proportional to the number of images in the image sequence, with the image-base fusion being the faster of the two, producing a 500x500 super-resolution image in a few seconds on a Ultrasparc.

If the variation of lighting is small, such as in an controlled indoor environment, the image-based approach is more appropriate because it uses the intensity information provided by the whole image sequence to construct the super-resolution image and thus is better at removing the noise and undesirable

$$B(x) = \begin{cases} \displaystyle\int_{x-\alpha}^{x+\alpha} vG(x-z)\,dz, & x < x_0 - \alpha \\ \displaystyle\int_{x-\alpha}^{x_0} vG(x-z)\,dz + \int_{x_0}^{x+\alpha} (v+\delta)G(x-z)\,dz, & x_0 - \alpha \le x \le x_0 + \alpha \\ \displaystyle\int_{x-\alpha}^{x+\alpha} (v+\delta)G(x-z)\,dz, & x > x_0 + \alpha \end{cases} \quad (1)$$

$$B(x) = \begin{cases} v\,\mathrm{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right), & x < x_0 - \alpha \\ (v + \dfrac{\delta}{2})\,\mathrm{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right) + \dfrac{\delta}{2}\,\mathrm{erf}\left(\dfrac{x - x_0}{\sqrt{2}\sigma}\right), & x_0 - \alpha \le x \le x_0 + \alpha \\ (v+\delta)\,\mathrm{erf}\left(\dfrac{\alpha}{\sqrt{2}\sigma}\right), & x > x_0 + \alpha \end{cases} \quad (2)$$

artifacts. On the other hand, the edge-based algorithm is more appropriate if the variation of illumination is large.

If the variation of lighting is intermediate, a possible solution is probably a hybrid of the two algorithms we propose herein. The idea is that instead of choosing a single reference image of the edge-based super-resolution algorithm, use the averaging or median of a sub-sequence out of the image sequence as the reference image, presuming that the variation of lighting is not so significant within the sub-sequence.

6 Future Work

Further work is needed before the super-resolution algorithm is robust enough for general use in VSAM applications; in particular, we need to incorporate more robust sub-pixel matching algorithms and include better deblurring algorithms. Quantitative analysis of both approaches is now under way using recognition rates as the benchmark.

7 Conclusion

This paper introduces two algorithms for enhancing image resolution from an image sequence. The image-based approach presumes that the images were taken under the same illumination conditions and uses the intensity information provided by the image sequence to construct the super-resolution image. The edge-based approach, based on edge models and a local blur estimate, circumvents the difficulties caused by lighting variations. We show that image warping techniques may have a strong impact on the quality of image resolution enhancement.

References

[Bascle *et al.*, 1996] B. Bascle, A. Blake and A. Zisserman. Motion deblurring and super-resolution from an image sequence. *Computer Vision—ECCV*, pages 573–581, Apr 1996.

[Boult and Wolberg, 1993] T. E. Boult and George Wolberg. Local image reconstruction and subpixel restoration algorithms. *CVGIP: Graphical Models and Image Processing*, 55(1):63–77, Jan 1993.

[Chiang and Boult, 1996a] Ming-Chao Chiang and T.E. Boult. Efficient image warping and super-resolution. *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 56–61, Dec 1996.

[Chiang and Boult, 1996b] Ming-Chao Chiang and T.E. Boult. The integrating resampler and efficient image warping. *Proceedings of the ARPA Image Understanding Workshop*, pages 843–849, Feb 1996.

[Chiang and Boult, 1997] Ming-Chao Chiang and T.E. Boult. Local blur estimation and super-resolution. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1997. To appear.

[Chiang, 1996] Ming-Chao Chiang. Imaging-consistent warping and super-resolution. Thesis Proposal, Columbia Univ., Dept of CS, Dec 1996.

[Gross, 1986] D. Gross. Super-resolution from subpixel shifted pictures. Master's thesis, Telaviv University, Oct 1986.

[Huang and Tsai, 1984] T. S. Huang and R. Y. Tsai. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, 1:317–339, 1984.

[Irani and Peleg, 1991] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53(3):231–239, May 1991.

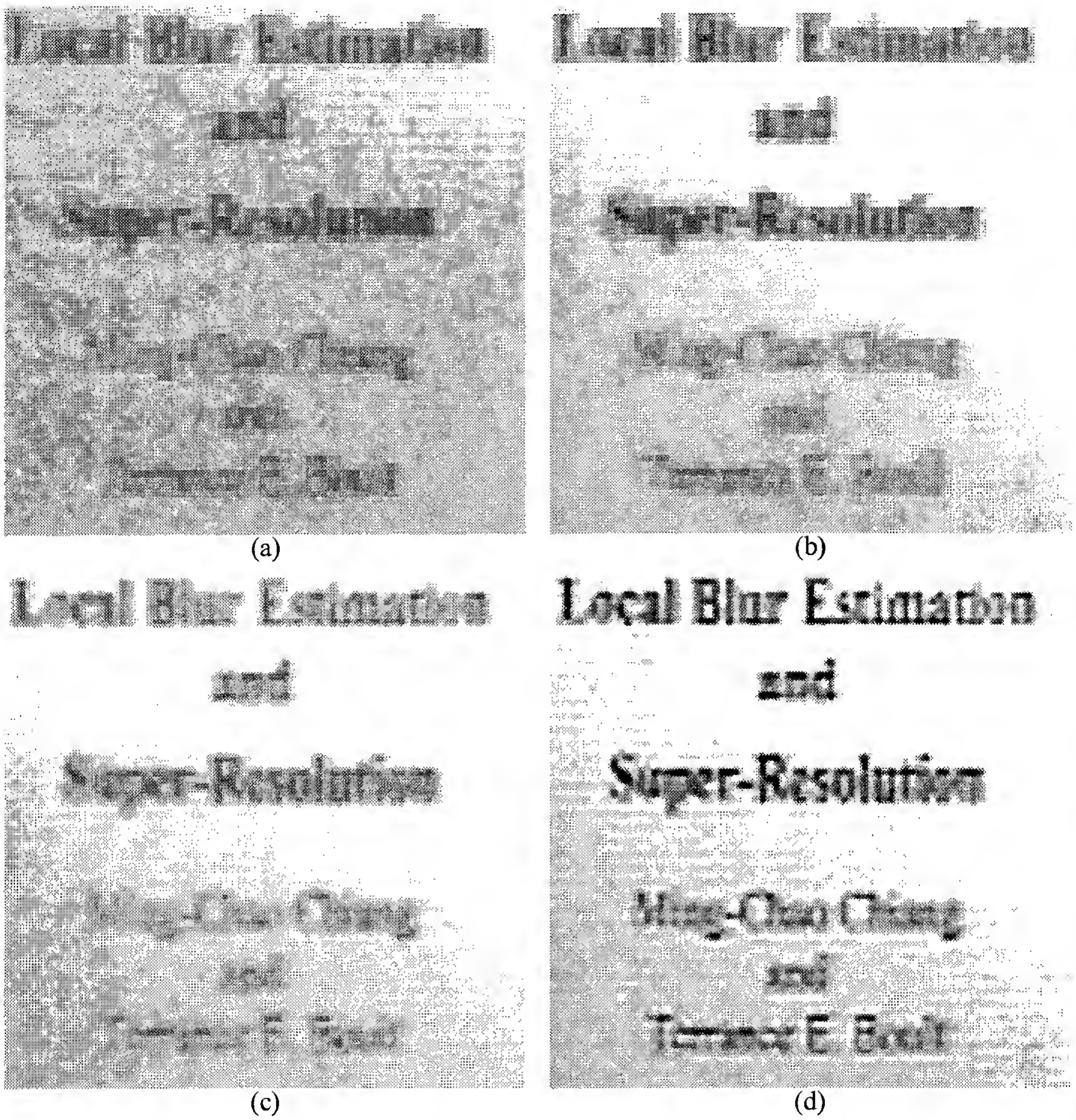


Figure 6: Edge-based example using 32 81x81 images: (a) and (b) are two of the original images blown up by a factor of 4 with pixel replication using pixel replication; (c) is super-resolution using the image-based algorithm without deblurring at the end; and (d) shows results with deblurring.

[Irani and Peleg, 1993] Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, Dec 1993.

[Keren *et al.*, 1988] D. Keren, S. Peleg and R. Brada. Image sequence enhancement using subpixel displacements. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, Jun 1988.

[Peleg *et al.*, 1987] S. Peleg, D. Keren and L. Schweitzer. Improve image resolution using subpixel motion. *Pattern Recognition Letter*, pages 223–226, 1987.

Multisensor Image Fusion Using a Region-Based Wavelet Transform Approach*

Zhong Zhang and Rick S. Blum
Electrical Engineering and Computer Science Department
Lehigh University, Bethlehem, PA 18015
zhz3@eecs.lehigh.edu rblum@eecs.lehigh.edu

Abstract

Using multiple sensors in a vision system can significantly reduce both human and machine errors in detection and recognition of objects. A particular case of interest is where images from possibly different types of sensors are to be combined. An image fusion scheme is proposed which combines aspects of feature-level fusion with the pixel-level fusion. Images are fused by combining their wavelet transforms. The identification of important features in each image, such as edges and regions of interest, are used to guide the fusion process. Experiments show that this algorithm works well in many situations.

1 Introduction

In recent years, multisensor image fusion has received significant attention for both military and industrial applications. Concealed weapon detection (CWD) is one interesting application. CWD appears to be a critical technology for dealing with terrorism. Detecting concealed weapons is especially difficult when one wants to monitor an area where portal systems are not practical. Portable systems, which could be placed in a police car, would be desirable. Due to the difficult nature of the problem, an extensive study indicated that no single sensor technology can provide acceptable performance over all of the scenarios of interest [Currie *et al.*-1996]. This justifies a study of fusion techniques to achieve improved CWD procedures. A number of compatible sensor technologies have already been identified which could provide improved performance if a fusion scheme were available [Currie *et al.*-1996]. Most of the technologies produce images, so image fusion is of interest.

We use the term image fusion to denote a process by which multiple images or information from multiple images are combined. These images may be obtained from different types of sensors. The majority of research on image fusion can be classified into the two categories: pixel-level image fusion and feature-level image fusion [Luo and Kay-95]. Pixel-level fusion generates a fused image in which each pixel is determined from a set of pixels in source images. The fused image is expected to be such that the performance of a particular task of interest, such as object detection, can be improved. Feature-level fusion first employs feature extraction separately on each image and then performs the fusion based on the extracted features. It enables the detection of useful features with higher confidence, and a fused image is not necessarily generated in this case. Currently, it appears that more people are focusing on pixel-level fusion.

Image fusion based on pyramid decomposition is one of the popular fusion methods. Pyramid decomposition methods construct a fused pyramid representation from the pyramid representations of the original images. The fused image is then obtained by taking an inverse pyramid transform. The approach was apparently first introduced in [Burt and Adelson-1983, Burt-1984] for image coding and binocular fusion in human vision. Several other pyramid-based image fusion schemes were proposed in [Toet-1990, Akerman III-1992, Burt and Lolczynski-1993]. More recently, approaches based on the wavelet transform have begun to receive considerable attention [Ranchin *et al.*-1993, Chipman *et al.*-1995, Li *et al.*-1995]. In [Ranchin *et al.*-1993], the authors studied fusion based on multiresolution

*This work is supported in part by ONR/DOD MURI program, contract N00014-95-1-0601. The source images in Fig. 3 were obtained from Thermotex Corporation.

image decomposition and reconstruction using the wavelet transform. They presented a technique for enhancing the spatial resolution of a SPOT image using another image from a different band from the same satellite. In [Chipman *et al.*-1995], the focus is on fusing multispectral aerial photos using a set of basic operations on particular sets of wavelet coefficients which correspond to certain frequency bands. In [Li *et al.*-1995], a wavelet transform approach is considered which uses an area-based maximum selection rule and a consistency verification step. The wavelet-transform-based approaches exhibit advantages in terms of compactness, directional selectivity and orthogonality [Li *et al.*-1995]. However, previous research had considered relatively simple methods for combining the wavelet coefficients which didn't make full use of the spacial information contained in the source images.

In this paper, we illustrate a wavelet transform based image fusion approach where we combine aspects of both pixel-level and feature-level fusion. The feature used is an object or region of interest which we refer to as a region here. Since objects and parts of objects carry the information of interest, it is reasonable to focus on them in the fusion algorithm. While previous researchers consider each pixel in the image separately, or just consider the pixel and its close neighbors, such as a 3 x 3 or 5 x 5 window, they neglect the fact that each pixel is just one part of an object or region. The objects are what we are really interested in. By considering each pixel separately, noise and blurring effects are often introduced during the fusion process. Our region-based method appears to be a good way to solve these problems.

In this paper, we consider fusion of two images only. Extensions to more than two images can be developed in a similar way. The proposed fusion scheme is described in Section 2. Some experimental results are presented in Section 3. Section 4 presents our conclusions.

2 The Region-Based Image Fusion Scheme

We take it as a prerequisite that the source images must be registered, so that the corresponding pixels are aligned. The discrete wavelet transform of each of the two registered images is computed. Then, using a scheme discussed later, the decision map is generated. Each pixel of the decision map denotes which image best describes this pixel. Based on the decision map, we fuse the two images in the wavelet transform domain. The final fused image is obtained by taking the inverse wavelet transform.

For each source image, the edge image, region image and region activity table are generated as shown in Figure 1. Next, the region activity tables of each image are used to create the fusion decision map. This is also illustrated in Figure 1. Each pixel in the fusion decision map tells which image should be used to provide the wavelet coefficients related to the corresponding pixel in the region image.

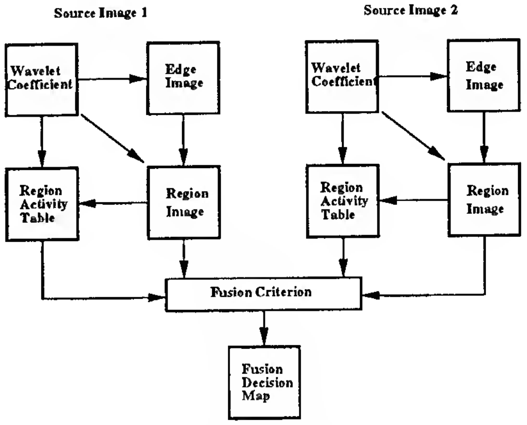


Figure 1: Data flow for creating the decision map

2.1 Wavelet Transform

The wavelet transform [Vetterli and Herley-1992, Mallat-1989] of an image provides a multiscale pyramid decomposition for the image. This decomposition will typically have several stages. There are four frequency bands after each decomposition. These are the low-low, low-high, high-low and high-high bands. The next stage of the decomposition process operates only on the low-low part of the previous result. This produces a pyramid hierarchy as shown in Figure 2, in which the top of the pyramid, denoted by LL^2, is a low-low frequency band. We can think of this low-low band as the lowpass filtered and subsampled source image. All the other bands which we call high frequency bands contain transform coefficients that reflect the differences between neighboring pixels and thus can be positive or negative. If we are dealing with grayscale images, then the absolute values of the high frequency coefficients represent the intensity of brightness fluctuation of the scene at a given scale. The larger values imply more distinct brightness changes which typically correspond to the salient features of objects. Thus, a simple fusion rule is to select the larger absolute value of the two corresponding wavelet coefficients from each of the two source images. There are two disadvantages of this method. It may have high sensitivity to noise and it may produce a blurring effect. To eliminate these undesirable features, we first divide each image into different objects and regions. Then, instead of performing the fusion pixel by pixel, we make the decision object by object and region by region. Thus in the fused image, each object will be described by the data from the clearer of the two images.

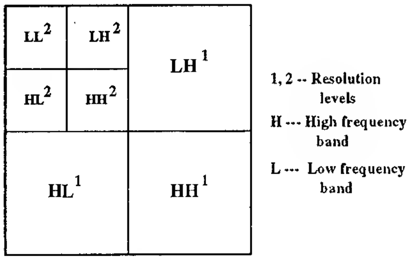


Figure 2: Pyramid hierarchy of wavelet transform

2.2 Region Labeling

We apply Canny edge detection to the low-low band of the coefficient pyramid obtained from the wavelet transform. The low-low band has a lower resolution than the source image, but it still contains the spacial region information. The output of the Canny detector is an edge image, on which region segmentation is performed using a labeling algorithm described in [Zhang and Blum-1997]. Finally, we get a labeled image in which each different value represents a different region, zero corresponds to edges.

The focus of this paper is on image fusion. While we have employed some specific edge detection and region labeling algorithms, other edge detection and region labeling algorithms, which may come from current or future studies, can be easily substituted for ours. The edge detection and labeling algorithms we choose are not necessarily the best. They just illustrate our approach.

2.3 Fusion

Information on salient features of an object is partially captured by the magnitude of high frequency wavelet coefficients that corresponding to that object. Consider two regions with similar size and signal-to-noise ratio (SNR) in two registered images which each represent the same object in a real scene. The one which has the larger magnitude high frequency components will generally contain more detail. Under this assumption, we first calculate the activity level of each region as the average of the absolute value of the high frequency band wavelet coefficients. Next we generate the decision map according to the activity level, size and position of each object or region. If the SNRs of two images are different, this can be accounted for by introducing a weight factor in the activity level calculation.

Recall that the region image corresponds to the low-low band of the wavelet coefficient pyramid. The activity level of region k in source image n Al_n(k) is given by

$$Al_n(k) = \frac{1}{N_k} \sum_{1 \le j \le N_k} P_j \qquad (1)$$

where N_k is the total number of pixels in region k, P_j is the activity intensity of pixel j in region k, which is given by

$$P_j = \frac{W}{M} \sum_{1 \le m \le M} \frac{1}{3 \cdot 2^{2(M-m)}} \sum_{1 \le i \le 3 \cdot 2^{2(M-m)}} |C_i| \qquad (2)$$

where W is the weight which is determined by the SNR of the image and other factors, M is the number of wavelet decomposition levels, C_i is one of the wavelet coefficients in high frequency bands corresponding to pixel j. The second sum in (2) is over all the wavelet coefficients that correspond to pixel j in the high frequency bands of the mth decomposition stage.

Next we describe how to produce the binary decision map. Suppose we have two registered images A and B to be fused. If a given pixel in the decision map is a "1" then all the wavelet coefficients corresponding to this pixel are taken from image A. If the pixel is "0", all the wavelet coefficients corresponding to this pixel are taken from image B. For a specific pixel of the decision map, P(i,j), this pixel may be:

1. in region m of image A, and region n of image B
2. an edge point in one image, and in certain region in the other image
3. an edge point in both images

We assign the value of each pixel in decision map according to the following criteria:

- Small regions preferred over large regions when comparing activity levels
- Edge points preferred over non-edges point when comparing activity levels
- High activity-level preferred over low activity-level
- Make decision on non-edge points first and consider their neighbors when making the decision on edge points
- Avoid isolated points in decision map

A binary decision map is now created to fuse the two wavelet coefficient arrays into one. Each pixel in the decision map corresponds to a set of wavelet coefficients in each frequency band of all decomposition levels. The size of the decision map is just $\frac{1}{2^M}$ of the original image where M is the number of decomposition stages. The value of M should not be too small, or we can not take the advantage of the decrease in image size due to the wavelet transform. In this case, the computation complexity will increase sharply. A large decomposition value is also not desired since resolution for region detection will be low. Practically, the choice of M is made according to the size of source image and its resolution. For our second example in Fig. 4, which uses 512 x 512 source images, an appropriate number of decomposition stages is two. In this case, the size of edge image, region image and decision map will be 128 x 128.

3 Experimental Result

We tested our algorithm on several pairs of images. Some of the result are described here. Figure 3 shows a pair of visual and 94GHz millimeter-wave (MMW) images. The visual image provides the outline and the appearance of the people while the MMW image shows the existence of a gun. From the fused image, we can clearly and promptly see that the person on the right has a concealed gun beneath his clothes. This fused image may be very helpful to a police officer, for example, who must response quickly. Figure 4 shows a pair of multifocused test images. ln one image, the focus is on the Pepsi can. ln the other image, the focus is on the testing card. In the fused image, the Pepsi can, the table, and the testing card are all in focus. These examples illustrate that our algorithm works in cases when the images either come from the same or different types of sensors.

4 Conclusion

We have presented a new approach to multi-sensor image fusion which combines the frequency information from wavelet transform with the spacial information from the original image. We use a particular image feature, regions which we believe represent objects, to guide the fusion process. Since objects and parts of objects carry the information of interest, it is reasonable to focus them in the fusion algorithm. Concealed weapon detection is one interesting application of our fusion algorithm. However, our algorithm can also be used in other applications.

References

[Akerman III, 1992] A. Akerman III. Pyramid techniques for multisensor fusion. In *Sensor Fusion V*, volume 1828, pages 124–131. SPIE, 1992.

[Burt and Adelson, 1983] P. J. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Communications*, Com-31(4):532–540, 1983.

[Burt and Lolczynski, 1993] P. J. Burt and R. J. Lolczynski. Enhanced image capture through fusion. In *Proc. the 4th Intl. Conf. on Computer Vision*, pages 173–182, Berlin,Germany, May 1993.

[Burt, 1984] P. J. Burt. The pyramid as structure for efficient computation. In *Multiresolution Image Processing and Analysis*, pages 6–35. Springer-Verlag, 1984.

[Chipman *et al.*, 1995] L.J. Chipman, T.M. Orr and L.N. Graham. Wavelets and image fusion. In *Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 208–219. SPIE, 1995.

[Currie *et al.*, 1996] N. Currie, F. Demma, D. Ferris, R. McMillan, M. Wicks, and K. Zyga. Imaging sensor fusion for concealed weapon detection. In *SPIE Symposium on Enabling Technologies for Law Enforcement and Security: Investigative Image Processing*. SPIE, Boston, MA, Nov. 1996.

[Li *et al.*, 1995] H. Li, B. S. Manjunath and S. K. Mitra. Multisensor image fusion using the wavelet transform. *Graphical Models and Image Processing*, 57(3):235–245, May 1995.

[Luo and Kay, 95] R.C. Luo and M.G. Kay. *Multisensor Integration and Fusion for Intelligent Machines and Systems*, pages 7–10. Ablex Publishing Corp, 95.

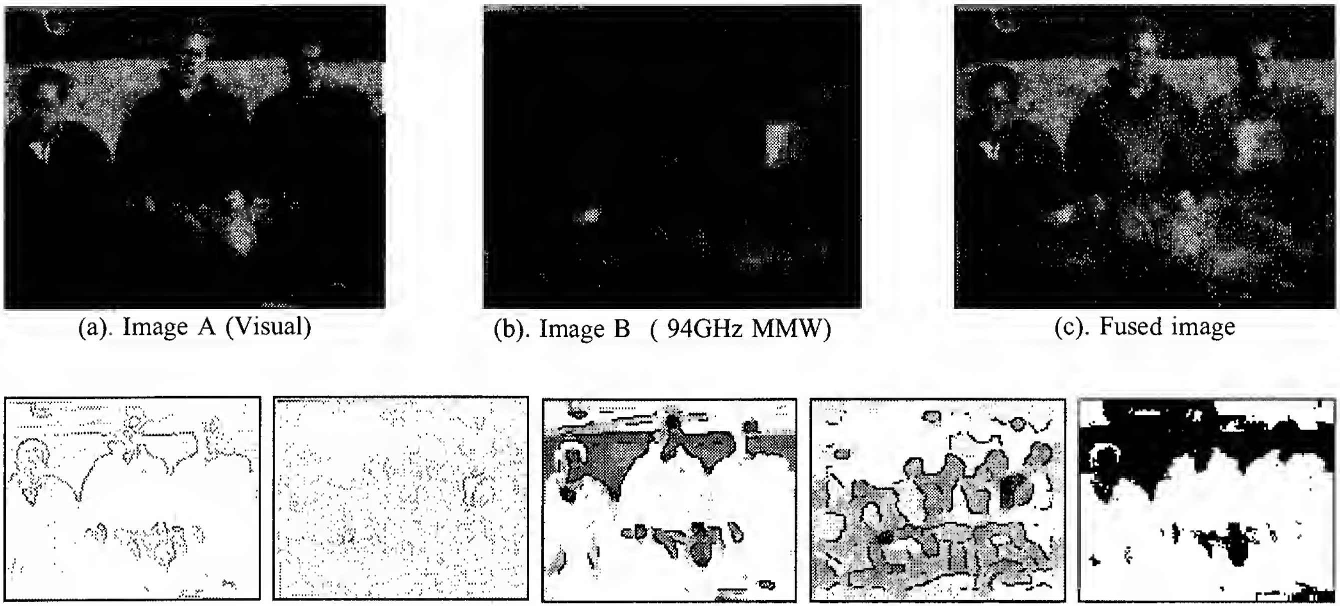
[Mallat, 1989] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-11:674–693, July 1989.

[Ranchin *et al.*, 1993] T. Ranchin, L. Wald and M. Mangolini. Efficient data fusion using wavelet transform: the case of spot satellite images. In *Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, volume 2934, pages 171–178. SPIE, 1993.

[Toet, 1990] A. Toet. Hierarchical image fusion. *Mach. Vision Appl*, pages 1–11, Mar 1990.

[Vetterli and Herley, 1992] M. Vetterli and C. Herley. Wavelets and filter banks: theory and design. *IEEE Trans. Signal Processing*, 40:2207–2232, September 1992.

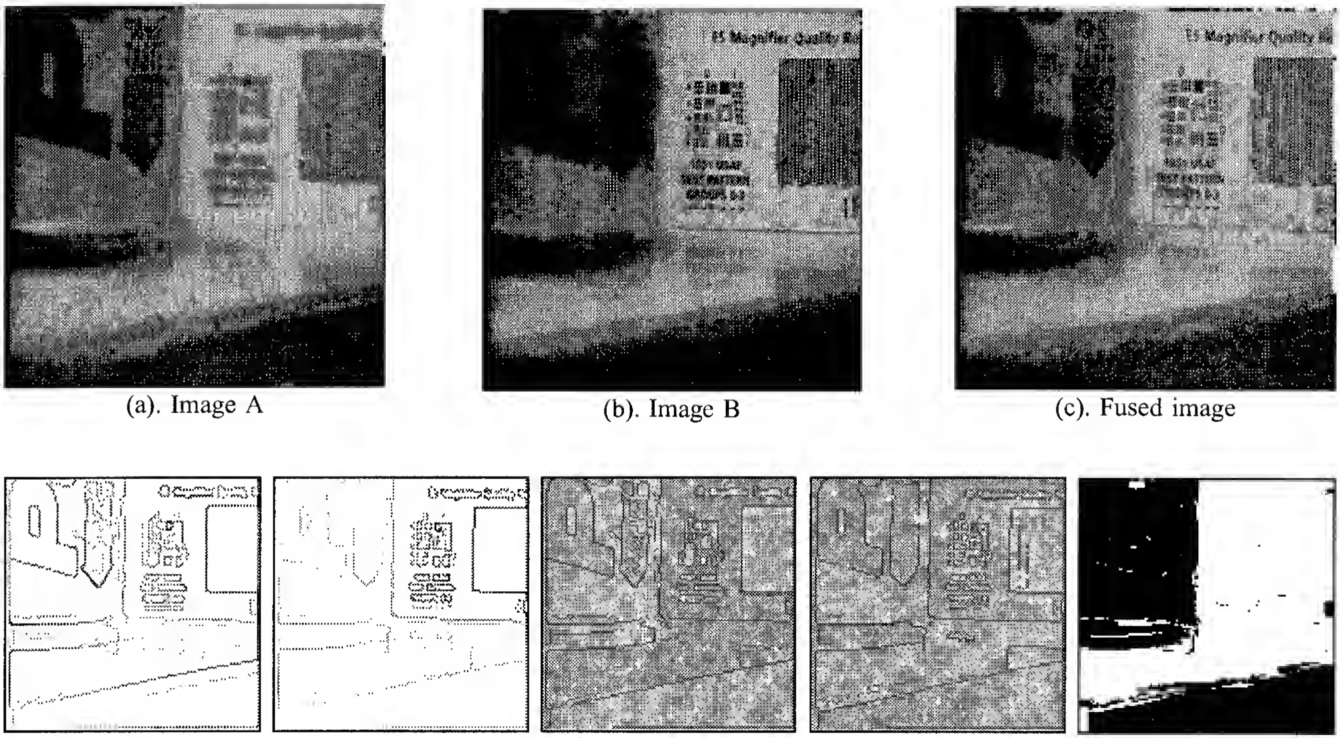
[Zhang and Blum, 1997] Z. Zhang and R. S. Blum. Region-based image fusion scheme for concealed weapon detection. In *Proc. 30th Conf. on CISS*, March 1997.



(a). Image A (Visual) (b). Image B (94GHz MMW) (c). Fused image

(d). Edge image A (e). Edge image B (f). Region image A (g). Region image B (h). Decision map

Figure 3: Fusion result on visual and radiometric images



(a). Image A (b). Image B (c). Fused image

(d). Edge image A (e). Edge image B (f). region image A (g). Region image B (h). Decision map

Figure 4: Fusion result on multi-focus images

A Variable Neighborhood Approach to Early Vision

Yuri Boykov | Olga Veksler | Ramin Zabih

Mathematics Department
Carnegie-Mellon University
Pittsburgh, PA
yura@andrew.cmu.edu

Computer Science Department
Cornell University
Ithaca, NY 14853
olga@cs.cornell.edu | rdz@cs.cornell.edu

Abstract

Early vision relies heavily on uniform, rectangular operators for tasks such as segmentation or computing correspondence. While rectangular windows have obvious efficiencies, they poorly model the boundaries of real-world objects. We have developed an efficient method for adaptively choosing a window without a rectangular bias, in a manner which varies at each pixel. We model an image as a piecewise constant function corrupted by noise, and explicitly consider all possible connected components. Almost all components can be pruned, however, by a simple maximum likelihood argument. The remaining components can be compared by a variety of methods, including (for example) global contextual constraints. Our approach can be applied to many problems, including image restoration, motion and stereo. It can help solve a number of well-known problems, including the aperture problem. Our methods run in a few seconds on traditional benchmark images with standard parameter settings, and give quite promising results.

1 Introduction

Many problems in early vision are ill-posed, i.e. they cannot be uniquely solved without additional constraints. Scene geometry can provide constraints that make these problems well-posed. For efficiency reasons, most algorithms make use of rectangular windows, which poorly model the boundaries of real-world objects. In this paper we present an approach to early vision problems that efficiently chooses a window adaptively at each pixel with no rectangular bias. The method can be used for image restoration, as well as for motion or stereo.

We will begin by looking at image restoration, and then generalize to motion and stereo. We model an image as a set of constant intensity regions (or connected components) that are corrupted by noise. Our approach has three parts. We first consider all possible intensities for all possible components. We will call the pair consisting of a component and an intensity a *component hypothesis.* A simple maximum likelihood argument prunes almost all the component hypotheses from consideration. The second part of our method is to rank the different component hypotheses. An easy way to do this is by preferring the largest components. The final part of our method is to assign each pixel in the image an intensity. This can be done locally, or with global consistency constraints.

2 Maximum likelihood hypothesis testing

Consider the problem of piecewise constant image restoration. Assume that $I_t(P)$ denotes the true intensity of a pixel P and that $I(P)$ denotes an observed intensity of the same pixel P after the picture is corrupted by noise, i.e. $I(P) = I_t(P) + \nu(P)$. If S is a connected set of pixels and i is an intensity, we will initially consider all possible component hypotheses. A component hypothesis S^i states that the pixels in the component S have the intensity i in I_t.

Let us pick a particular intensity $\hat{\imath}$. The key step in our method is that a pixel P is included in $S^{\hat{\imath}}$ if and only if P is more likely to have intensity $\hat{\imath}$ than not. If every pixel in $S^{\hat{\imath}}$ has this property, then we will call the component hypothesis $S^{\hat{\imath}}$ *plausible*. By only considering plausible component hypotheses, we can enormously reduce the amount of work.

To enforce plausibility we must decide between the following two hypotheses:

$$\begin{aligned} H_0 &: I_t(P) = \hat{\imath}, \\ H_1 &: I_t(P) \neq \hat{\imath}. \end{aligned}$$

Assume that the noise model is given by the function

$$f(x|i) = \Pr(\, I(P) = x \mid I_t(P) = i\,). \qquad (1)$$

This function may correspond to a uniform, Gaussian, or any other distribution function. We will write $f(x|i)$ as $f(I(P)|i)$ where $I(P)$ will serve as an intensity value observed in a fixed experiment (not as a random variable).

We choose between H_0 and H_1 by comparing the likelihoods $\Pr(I(P)|H_0)$ and $\Pr(I(P)|H_1)$; in other words, we assume there is no prior bias in favor of H_0 or H_1. The hypothesis with the largest likelihood wins. Obviously, we have

$$\Pr(I(P)|H_0) = f(I(P)|\hat{\imath}).$$

To compute $\Pr(I(P)|H_1)$ we proceed as follows:

$$\begin{aligned} \Pr(I(P)|H_1) &= \frac{\Pr(I(P), H_1)}{\Pr(H_1)} \\ &= \sum_{i\neq\hat{\imath}} \frac{\Pr(I(P), I_t(P) = i)}{\Pr(H_1)} \\ &= \sum_{i\neq\hat{\imath}} \frac{f(I(P)|i)\Pr(I_t(P) = i)}{\Pr(H_1)}. \end{aligned}$$

We assume that prior probabilities of all intensities are equal. This implies that $\Pr(I_t(P) = i)$ does not depend on i. Thus,

$$\Pr(H_1) = (|R| - 1) \cdot \Pr(I_t(P) = i), \qquad \forall i$$

where $|R|$ denotes the number of all possible intensities. Therefore

$$\Pr(I(P)|H_1) = \frac{1}{(|R| - 1)} \cdot \sum_{i\neq\hat{\imath}} f(I(P)|i).$$

To choose H_0 over H_1 we require

$$f(I(P)|\,\hat{\imath}\,) \geq \frac{1}{(|R| - 1)} \cdot \sum_{i\neq\hat{\imath}} f(I(P)|\,i\,).$$

Finally, this comparison test can be rewritten as

$$f(I(P)|\,\hat{\imath}\,) \geq \frac{1}{|R|} \cdot \sum_{i} f(I(P)|\,i\,), \qquad (2)$$

so that we accept hypothesis H_0 in cases where the likelihood of intensity $\hat{\imath}$ is larger than or equal to the *average likelihood* of all possible intensities.

At this point it becomes obvious that we can use any noise model $f(\,\cdot\,|i)$ in formula (2). If $f(\,\cdot\,|i)$ is uniform, Gaussian, or any other non-increasing symmetric distribution function centered at i then test (2) can be reformulated as

$$|I(P) - \hat{\imath}| \leq \epsilon \qquad (3)$$

where ϵ is a parameter that depends on the particular distribution function $f(\,\cdot\,|i)$ in hand. Note that the threshold ϵ is the same for all pixels P in the image. This furnishes a computationally trivial way to compute the plausible component hypotheses.

2.1 Motion or stereo

Consider now the problem of motion of stereo. We will denote a disparity by d. We will write the statement that pixel P has disparity d by P^d. $I(P)$ and $I'(P)$ will represent intensities of pixel P in the first and in the second images, respectively. Consider some fixed disparity $\hat{d}$ for pixel P. Again, we will enforce plausibility on every component hypothesis. We need to choose between the two hypotheses:

$$\begin{aligned} H_0 &: P^{\hat{d}} \\ H_1 &: \neg P^{\hat{d}}, \end{aligned}$$

Assume that the function $f(\,\cdot \mid i'\,)$ specifies the noise model, that is the distribution of intensity of a pixel in the first image given intensity

i' of the corresponding pixel in the second image. For convenience, we define for any event E $\Pr'(E) = \Pr(E|I'(P_1), \ldots, I'(P_n)) = \Pr(E|I')$, which is the probability of E conditioned on all the observed intensities from the image I'. Similarly we define $\Pr'(E|F) = \Pr(E|F, I')$.

Then

$$\Pr'(\, I(P) \mid P^d \,) = f(\, I(P) \mid I'(P+d)\,)$$

where $P+d$ is a pixel obtained by shifting P by disparity d.

We choose between H_0 and H_1 by comparing the likelihoods $\Pr'(I(P)|H_0)$ and $\Pr'(I(P)|H_1)$. Obviously, we have

$$\Pr'(I(P)|H_0) = f(I(P)|I'(P+\hat{d})).$$

To estimate $\Pr'(I(P)|H_1)$ we proceed as follows:

$$\begin{aligned}\Pr'(I(P)|H_1) &= \frac{\Pr'(I(P), H_1)}{\Pr'(H_1)} \\ &= \sum_{d \neq \hat{d}} \frac{\Pr'(I(P), P^d)}{\Pr'(H_1)} \\ &= \sum_{d \neq \hat{d}} \frac{f(I(P)|I'(P+d))\Pr'(P^d)}{\Pr'(H_1)}.\end{aligned}$$

We assume that prior probabilities of all disparities are equal. This implies that $\Pr'(P^d)$ does not depend on d. Consequently,

$$\Pr'(H_1) = (|D|-1) \cdot \Pr'(P^d), \qquad \forall d$$

where $|D|$ denotes the number of all possible disparities. Therefore

$$\Pr'(I(P)|H_1) = \frac{1}{(|D|-1)} \cdot \sum_{d \neq \hat{d}} f(I(P)|I'(P+d)).$$

To prefer H_0 over H_1 we should have

$$f(I(P)|I'(P+\hat{d})) \geq \frac{1}{(|D|-1)} \cdot \sum_{d \neq \hat{d}} f(I(P)|I'(P+d)).$$

Finally, this comparison test can be equivalently rewritten as

$$f(I(P)|I'(P+\hat{d})) \geq \frac{1}{|D|} \cdot \sum_{d} f(I(P)|I'(P+d)), \tag{4}$$

so that we accept hypothesis H_0 in cases where the likelihood of disparity $\hat{d}$ is larger than or equal to the *average likelihood* of all possible disparities.

We can use any noise model $f(\,\cdot \mid i'\,)$ in formula (4). If f satisfies[1] $f(\,x \mid i'\,) = f(x - i')$ and if ΔP^d denotes $I(P) - I'(P+d)$ then test (4) is equivalent to

$$f(\Delta P^{\hat{d}}) \geq \frac{1}{|D|} \cdot \sum_{d} f(\Delta P^d). \tag{5}$$

This test is equivalent to

$$|\Delta P^{\hat{d}}| \leq \epsilon$$

where ϵ depends on the noise model f. Again, this provides a computationally trivial way to compute the plausible component hypotheses.

2.2 Comparing component hypotheses and producing pixel estimates

The results above show that instead of considering all possible component hypotheses we only need consider a small set. In particular, for a particular hypothesis (intensity or disparity), we only consider plausible components, i.e. components of pixels whose likelihood under that hypothesis is above the pixel's average likelihood for all hypotheses. There are typically a small number of plausible component hypotheses.

The next step is to rank the different component hypotheses according to some preference criterion. The criterion should combine two sources of information: geometric information, encoded as priors, as to which segmentations are most likely; and statistical information regarding the residuals (i.e. ΔP^d) within the component. In most of our experiments so far we have used the simple criterion that the largest component is the best explanation. However, more complex priors may also be used, and contextual information can be incorporated. For example, with stereo it is often desirable to find a ground plane

[1] This holds if $f(\,\cdot \mid i'\,)$ is uniform, Gaussian, or any other (bell-shaped) symmetric distribution function centered at i'.

somewhere in the lower part of the image; this constraint can be incorporated into the preference criterion. Statistical information about the residuals can also be used; at the correct match, the residuals should come from measurement noise, and should (for example) be unbiased.

Once the component hypotheses have been ranked, the final step is to assign a value (intensity or disparity) to each pixel. The simplest solution is to do this purely locally. For each pixel P, we consider those component hypotheses containing P. The best component hypothesis, under the preference criterion, will be used to give P its value. To date, we have mostly used this local method, but more complex methods may eventually be desirable. For example, it may be important to impose a kind of global consistency, so that every pixel in the best component hypotheses is ensured the same value.

2.3 Efficiency

The computation of the plausible component hypotheses can be done in linear time, in a single pass over the images. The methods we have used for comparing component hypotheses, and for assigning values to pixels, are similarly efficient. In practice, our initial implementation takes a few seconds per image to compute depth. For example, the stereo images shown in section 3 took 3 seconds to process with 10 disparities, on a 50-MHz sparc.

3 Experimental results

We have used our methods both for image restoration and for stereo and motion. For image restoration, we took a synthetic image of a set of inscribed diamonds and corrupted it with gaussian noise with $\sigma = 6$. The original intensities are arranged with gaps of 40 intensities between adjacent regions. The restored image is shown in figure 1. 98% of the pixels were assigned the correct intensity by our method. For this example we used $\epsilon = 18$.

For the stereo experiments presented below we used $\epsilon = 2$ to account mostly for the camera noise. To handle other measurement errors we introduced gain and bias parameters g, b that adjust pixel intensities in the right image. Our model of image formation becomes

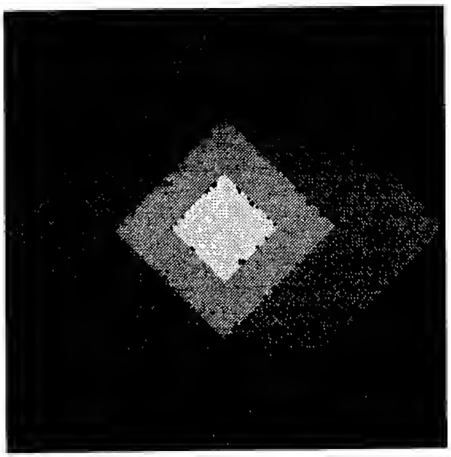


Figure 1: Image restoration of a synthetic image

$$I(P) = (g \times I'(P + d) + b) + \nu(P). \quad (6)$$

Connected components were computed for all disparities in D and for all values of adjustment factors in a fixed range. For all real images shown here, we varied g from 0.8 to 1.2 and b from -18 to 18. Finally, for each pixel P we chose disparity, g, and b corresponding to the largest connected component containing P. This is a simple way to deal with g and b parameters.

There are usually a number of pixels in the left image which are occluded in the right image, and the attempt to find a match for such pixels will yield wrong answers. There is an easy way to cut down the amount of such errors our algorithm produces. Suppose for some pixel P in the left image, the size of the largest connected component P is consistent with is smaller than some fixed minimal size. Then we consider pixel P to have no match in the right image. We fixed the minimal size component to be 6 for all the result shown in this section.

We also check the output disparity for double assignments. That is if two pixels in the left image are mapped to the same pixel in the right image, then the pixel which belongs to a bigger connected component gets assigned to that pixel in the right image.

Figure 2 shows the left image of a random dot stereogram. In the right image, the entire image in translated uniformly. Note the large textureless region in the center, which creates problems for existing algorithms. Our method performs correctly for this image, assigning all pixels the correct translation. Our adaptive window scheme ensures that the entire image is treated as a single window, thus propagating information from the high-texture regions at the outskirts of the image into the low-texture regions in the middle of the image. Kanade and Okotumi's algorithm [Kanade and Okutomi, 1994] may also succeed in this situation.

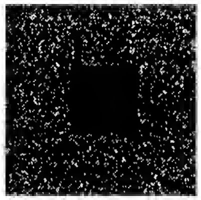


Figure 2: Random dot stereogram illustrating aperture problem

3.1 Results on Real Images

Figure 3 shows the left image of the meter pair from CMU and the results of normalized correlation and our algorithm. In this image on the wall of the building there are many areas of low intensity variation. Normalized correlation clearly has trouble producing correct answers in these areas. Even if we use a large window of 20 × 20 pixels, there are still a lot of large bright spots on the wall, which were obviously not matched correctly. Our algorithm places most of the wall at the same disparity. We don't know if it is indeed the correct answer, but normalized correlation not only places most pixels of the wall at this disparity, but it also places large patches of pixels at this particular disparity at the right and the left ends of the wall.

It's interesting to observe how the algorithm works for the car. Almost all of the car except for very few pixels were placed at the same disparity. The edge of the car produced by our algorithm is very sharp.

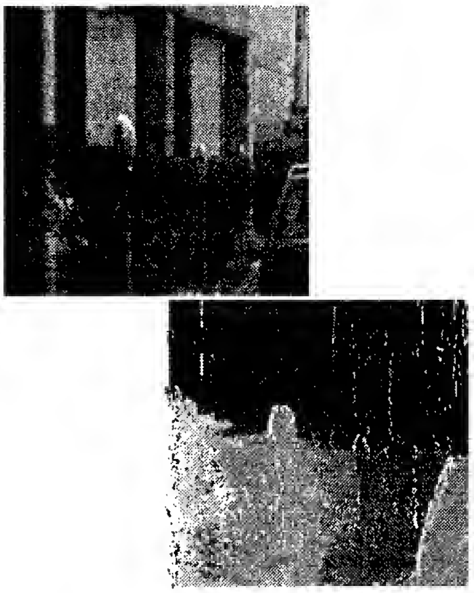


Image Our results

Figure 3: Meter results

References

[Kanade and Okutomi, 1994] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

Toward Automatic Domain-Adaptation in Artificial Evolution: Experiments with Face Recognition

Matthew R. Glickman* and **Katia P. Sycara***
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15381-3891 USA
E-MAIL: glickman+@cs.cmu.edu,katia@ri.cmu.edu

Abstract

Organisms found in nature exhibit highly adaptable, robust visual abilities, hinting that biologically inspired methods such as artificial evolution are a potentially fruitful path toward inducing such abilities in artificial systems. However, the performance of search in any given domain is often highly dependent on the amount of knowledge of the particular domain that is applied to guide search. This fact poses multiple problems for artificial evolution in visual task domains. At the same time, natural evolution has succeeded despite these obstacles, suggesting that it may be possible to obtain useful knowledge of the domain in question over the course of evolution. The experiments reported here, conducted in the domain of artificial neural networks for face recognition, demonstrate that useful domain knowledge generated as a by-product of evolution may be extracted, analyzed, and used to improve further search in the domain.

1 Introduction

Search algorithms patterned after biological evolution are attractive for use in domains such as vision that have complex search spaces for a number of reasons, including: (1) Application of them doesn't explicitly require deep insight into the domain; (2) they're relatively straightforward to parallelize; and (3) their natural analog has resulted in entities of extraordinary complexity and robustness. However, the performance of search in any particular domain is highly dependent on the interaction between the chosen representation of the space and the specific search operators employed, and for evolutionary algorithms in particular, this interaction is a poorly understood process, leaving practitioners with few guidelines as to how to make the right choices to yield good performance.

One popular approach to improving the performance of search in a particular domain is to seek to incorporate pre-existing knowledge of the domain into the operators and representation. However, this approach is problematic for evolutionary search because of the aforementioned opacity of the interaction between the operators and the representation. This difficulty, popularly known as "the representation problem", is only compounded in more complex domains, presenting a formidable obstacle to the application of artificial evolution in precisely those domains in which they may be of the greatest utility.

Moreover, even if knowledge useful for specific visual tasks of interest is available, "hard-coding" any available domain knowledge into artificial evolution may actually inhibit adaptation when domain parameters change. There-

*This research was partly sponsored by Office of Naval Research (ONR) under Contract N00014-95-1-0591. Matthew Glickman was supported in part by an NSF Graduate Fellowship Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR, NSF, or the U.S. Government.

fore, rather than seeking how pre-existing domain knowledge can be best exploited by evolution, our research is directed toward the automatic acquisition of such knowledge in operational form. We have developed *domain-independent* techniques for learning *domain-adapted* control knowledge that guides evolutionary search.

The experiments reported herein demonstrate that generalized information about a particular domain, generated over the course of evolutionary search, can be extracted, analyzed, and then employed to improve search in future runs. Our application, automated face recognition, is an instance of the general *signal-to-symbol mapping problem* [Teller and Veloso, 1995]. In particular, the task is to identify which of five individuals is represented in a given visual image.

The space explored is the weight space of fixed-topology, feed-forward artificial neural networks (ANN's). Over the course of adaptation, weight vectors were collected along with their self-adapted, variable mutation rate (*e.g.* [Schwefel, 1995], [Fogel *et al.*, 1991]). These data were then used to train another ANN to predict the appropriate mutation rate for a given weight vector for the face-recognition domain in general. Finally the mutation rate-prediction networks were used to drive evolution on another face recognition task, resulting in networks with improved generalization performance.

2 The Face Recognition Domain

The image set (based on data and code provided by Jeff Shufelt and Tom Mitchell, available at `http://www.cs.cmu.edu/~tom/faces.html`) consists of approximately 8 gray-scale images of each of 20 people. Each person is pictured facing forward, from the shoulders up, making a variety of facial expressions, both with sunglasses and without.

The images were scaled to 30x32 pixels of one byte each, yielding a total of 960 bytes which are then provided as input to a feed-forward artificial neural network with a single hidden layer and an output layer, each consisting of five sigmoid units.

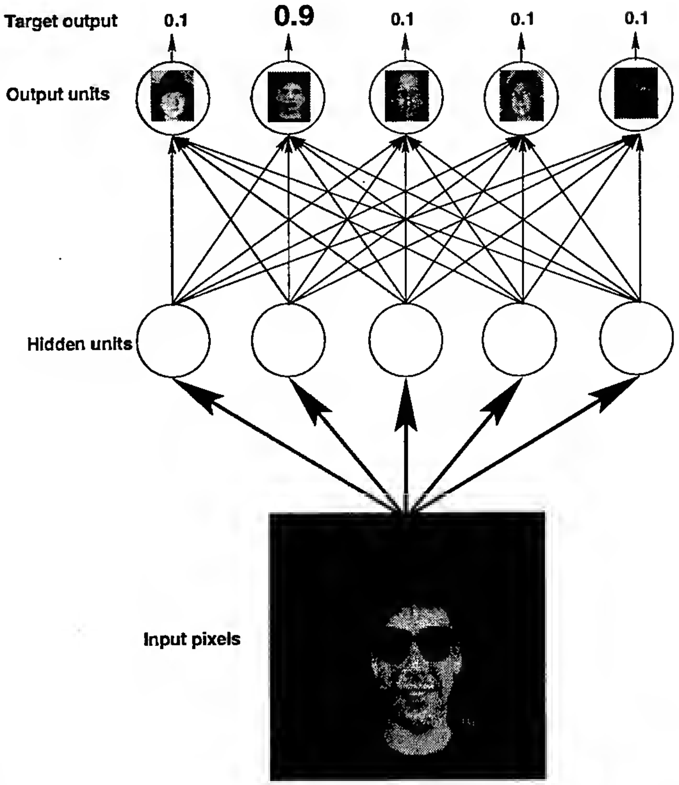


Figure 1: Face recognition artificial neural network

We divided the image set of the 20 people into four *task sets*, $S1$ - $S4$, of five people each. Each task set was further divided into training and testing sets, with 5 images of each individual allocated to training, and 2 or usually 3 to testing. For each task set, a single output unit was designated as a "recognizer" for each person in the set. When presented with an image of a given person, the target output for the network was 0.9 from the recognizer output unit, and 0.1 from the rest (see figure 1).

3 Evolving Face Recognition Networks

Artificial evolution, typified by such paradigms as Evolutionary Programming ([Fogel *et al.*, 1966],[Fogel, 1995]), Evolution Strategies ([Rechenberg, 1973],[Schwefel, 1995]), and Genetic Algorithms ([Holland, 1975],[Goldberg, 1989]), may be viewed as the alternating, iterative application of two processes to a population of candidate objects:

- the *selection* process transforms the population to one in which the most promising candidates tend to be represented in greater frequency, thus boosting the mean quality of the population
- the *variation* process then stochastically perturbs individuals in the population in some manner, with the hope of producing individuals of higher quality than any yet found.

The specific instantiation of artificial evolution employed here most closely matches that of the Evolution Strategies paradigm, in which the principal source of variation is Gaussian mutation, implemented with self-adapting, variable mutation rates.

3.1 Phase 1 - Collection of Domain Knowledge

In the first phase of our experiments, several runs of the evolutionary algorithm were conducted, each of which adapted *face recognition networks* (FRN's) to perform the recognition task for one of the task sets $S1$, $S2$, or $S3$.

Each run iteratively transformed a population of 100 individual FRN's. Each FRN i was represented as a vector of real values, with one value for each weight in the network, $w_{i1} \ldots w_{in}$, and one additional value for the *mutation rate* (see below), m_i. In the initial population, all the weight values were initialized to 0, and the mutation rates were set to 0.1.

3.1.1 The Selection Process

The selection process presented each FRN with each of the training images from the target task set for the run in question as input, and assigned each network a *fitness* value equal to its sum of the squared error (SSE) with respect to each of the target output vectors (0.9 for the output unit assigned to recognize the person depicted in the input image, and 0.1 for the rest) for each image. A new population–with a lower mean SSE–was then made by copying each of the 10 FRN's with the lowest SSE on the training images 10 times. This form of selection is known as (μ, λ)-*selection*, where μ (10 in this case) is the number of "parents" chosen to reproduce to form a new population of size λ (100 in this case).

3.1.2 The Variation Process

In the variation process, each value in each weight vector (each vector representing a FRN) was mutated using Gaussian noise. As is typical of the Evolution Strategies paradigm, the mutation rate values were each multiplied by a value sampled from an exponential Gaussian distribution with a mean of 0 and a standard deviation of 0.1. For each individual FRN, its resulting mutation rate was then used as the standard deviation of another Gaussian with a mean of 0. For each weight value of the FRN in question, a value was sampled from this distribution and added to the weight:

$$\begin{aligned} m_i' &= m_i \cdot exp(Gaussian(\mu = 0, \sigma = 0.1) \\ w_{ij}' &= w_{ij} + Gaussian(\mu = 0, \sigma = m_i'), \\ &\quad \text{for } j = 1 \cdots n \end{aligned}$$

Evolutionary algorithms such as this one have previously been shown to produce good results for training ANN's (*e.g.* [Porto *et al.*, 1995], [Baluja, 1995], [Glickman and Sycara, 1996]). The minimum SSE in the population over time for both the training and test sets (SSE was measured for the test set to observe the change of generalization performance over time, but only SSE on the training set was used for the purposes of the selection process*) is shown for a typical run in figure 2.

Training was stopped after 500 generations, both for purposes of expediency and because we determined that adaptation had typically slowed to a relatively insignificant rate by this point. Moreover, in the great majority of cases (including the run shown in figure 2), despite the fact that the SSE with respect to the target output vectors was still above zero, by 500

*The SSE depicted for the test set at each generation is for the same FRN for which the SSE on the training set is shown, *i.e.* the one in the population with the lowest SSE on the training set. This FRN is thus not necessarily the one with strictly the lowest SSE on the test set in the population.

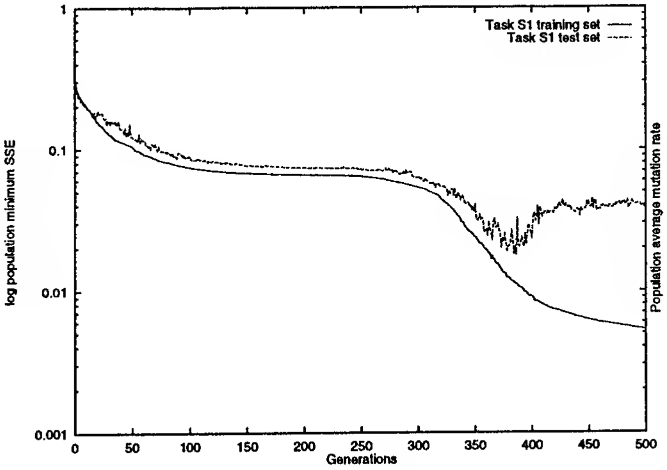


Figure 2: A typical run on task set S1

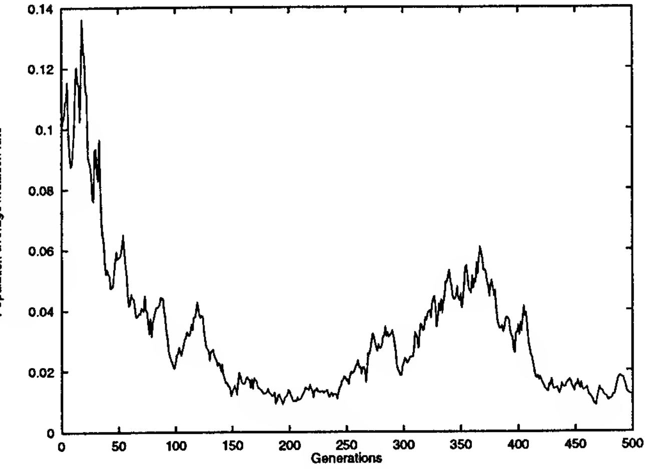


Figure 3: Adaptation of mutation rates for a typical run on task set S1

generations all of the images in both training and test sets were found to have long since been properly classified (with classification being defined as the designated output unit producing a value above 0.5, while the others producing values below 0.5).

3.1.3 Self-adapting mutation rates

As stated above, the mutation rates for each FRN are themselves subject to mutation. Despite the fact that the mutation rates are not directly measured by the fitness function, selection results in their adaptation as well, via an indirect mechanism called *self-adaptation* in the Evolutionary Computation literature.

A FRN that is selected for the next generation has 10 "offspring". Each of its offspring inherits its parent's mutation rate, which is then mutated, and then used to mutate its weights. Selection is thus seen to work on mutation rates via favoring those individuals with mutation rates that resulted in a favorable mutation to their own weights. Given two different networks, the "best" mutation rate– the one most likely to produce a favorable mutation in one network–may not be the same as the "best" mutation rate for the other. Selection will favor them depending how well their mutation rates are adapted to their own specific position in weight space, hence the "self" in self-adaptation.

At the same time, it is a general rule that the better adapted a particular individual is–in this case, the lower the SSE on the training set–the more likely mutation is to degrade its fitness, and the higher the mutation rate, the greater the expected extent of degradation. Thus, increasingly lower mutation rates *tend* to be favored overall as adaptation proceeds (although not exclusively) as is shown in figure 3 (for the same sample run as in figure 2).

Figure 3 also illustrates two other important points. One is that selection on mutation rates is noisier than that on SSE. This is to be expected because the rates themselves are not directly measured by selection, and because even a poor rate has a chance of producing good offspring while a good rate can produce poor offspring. Another observation to make is that while the *general* tendency is for mutation rates to decrease, this tendency is not necessarily monotonic. The average population mutation rate in figure 3 is actually seen to rise for a period lasting from approximately generations 200 - 375. During this same period, figure 2 shows that the population minimum SSE on the training images dropped significantly after a period of relative stagnation. One may therefore infer that during this period the networks in the population discovered a region of the weight space in which higher mutation rates enabled them to adapt more rapidly. The purpose of noting this occurrence is to draw attention to the fact that while the overall trend is for mutation rates to decay over time (*e.g.* see the line depicting self-adapting mutation rates averaged over several runs in figure 5 below), at any particular point in any particular run, self-adapting mutation rates are sensitive to the local contours of the fitness surface.

Thus, the mutation rates appear as clues to a sort of noisy map of the fitness surface. However, because the rates do decay as a local optimum is approached, the information isn't usually retained at the end of the run. To see whether generalized domain information might be extracted from these traces, we collected a set, Σ, of the best weight vectors from each generation along with their associated mutation rates over several runs on each of task sets $S1$, $S2$, and $S3$. In the next phase of these experiments, we attempted to use these data to learn to predict, given a weight-vector representing a FRN, a mutation rate that would be suitable for mutating this vector.

3.2 Phase 2 - Analysis of Domain Knowledge

In the second phase, we again trained ANN's. This time, however, the networks trained were *mutation rate prediction networks* (MRPN's), mapping FRN weight-vectors to appropriate mutation rates. The MRPN'S were assigned three hidden units, and trained using backpropagation for speed. It is important to point out that our goal was simply to determine whether these data were in fact learnable to any useful extent[†].

After periodic sampling to reduce the volume of training examples to a reasonable number, we allocated approximately two-thirds of the dataset Σ collected in phase 1 to training, and the remaining one-third to testing, taking care that the data allocated to training and testing were from *separate* runs in order to avoid overfitting particular runs. The weight vectors were then presented as input and their associated mutation rates were used as the target output. We stopped training after overfitting began to occur, and saved the MRPN's with the lowest error on the test data for use in the third phase of the experiment.

After training, the MRPN's had learned to predict, for a given FRN weight vector, a mutation rate at which it would be advantageous to mutate the given vector in order to achieve adaptation. Because the MRPN's were trained from data collected from a variety of runs conducted on multiple different task sets, to the extent to which an MRPN was successful at guiding search on *another* face recognition task set on which it hadn't been trained, it could be said to embody some form of knowledge about the FRN domain in general. In phase 3, we conducted experiments to determine whether this was in fact the case.

3.3 Phase 3 - Use of Domain Knowledge

In the final phase, face recognition networks were once again evolved, but this time using task set $S4$. This time, instead of mutating the selected weight vectors with an associated mutation rate, each selected vector was provided as input to one of the MRPN's trained in phase 2. The resulting mutation rate was then used to mutate the connection weights:

$$\begin{aligned} m_i &= MRPN(\vec{w}_i) \\ w'_{ij} &= w_{ij} + Gaussian(\mu = 0, \sigma = m_i), \\ &\quad \text{for } j = 1 \cdots n \end{aligned}$$

The mutation rates in these runs were therefore no longer being self-adapted, but rather determined by the MRPN's learned in phase 2. For the purposes of comparison, a set of runs with self-adapting rates, analogous to those conducted in phase 1, were also conducted on $S4$.

4 Results

Figure 4 shows the log minimum population SSE on the task $S4$ test set, averaged by generation, over the first five runs achieved using two separate MRPN's trained in phase 2 to determine the mutation rates. Average results

[†] The choice of ANN's for this phase was essentially arbitrary; any one of a host of other function approximators may quite likely yield better performance for the mutation rate prediction task.

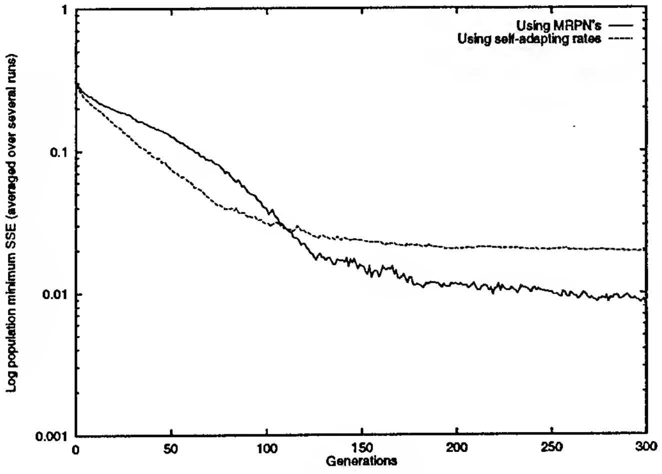


Figure 4: Average log population minimum SSE on the task $S4$ test set over time

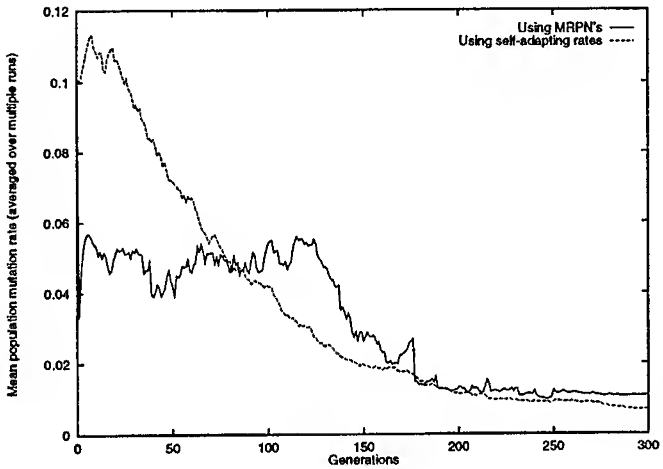


Figure 5: Average population mutation rate on task $S4$ over time

for runs using self-adapting mutation rates conducted on task S4, analogous to those conducted in phase 1, are shown for comparison. Use of the MRPN's over these initial runs showed a non-trivial improvement in generalization performance on the face recognition task as measured by performance on the test images. Some measure of insight as to the source of this improvement may be gleaned from figure 5.

Most significantly, note that the population average mutation rate is indeed attenuated as adaptation proceeds, signifying that the MRPN's *are* indeed managing to recognize FRN vectors that are more adapted, and specify a lower mutation rate as is appropriate. Moreover, because the MRPN's were trained on data collected from runs adapting FRN's on task sets other than the ones examined here, their mutation rate "recommendations" must be being made based on generalized FRN domain knowledge, rather than specific knowledge of task set $S4$.

Despite the fact that some domain knowledge has apparently been acquired, figure 5 does show a substantial difference between the average rates specified by the MRPN's and those found via self-adaptation. If the self-adapted rates are in fact well-adapted, why do the differing rates specified by the MRPN's result in an advantage (*i.e.* lower SSE) with respect to generalization performance (figure 4). The answer is unknown at this time, however one hypothesis is that a run employing self-adapting rates is spending a significant amount of its search capacity per generation exploring the space of mutation rates. In other words, with self-adapting rates, FRN's are continually being spawned with mutation rates that are either too high or too low to result in a reasonable rate of adaptation. In runs employing the MRPN's, mutation rates are determined in a much more deterministic manner, focusing adaptation on the FRN's rather than on their mutation rates themselves.

5 Conclusion

Many related questions remain to be answered. In particular, the question of how to best approximate the function mapping weight vectors to mutation rates for this domain remains wide open. ANN's and backpropagation were used in this case, but we know of no reason to expect them to perform better than any other function approximator. Moreover, MRPN's trained on differing samples of the phase 1 data differ in their ability to control subsequent searches. There clearly remains a great deal of exploration to do with respect to both how to sample the data collected in phase 1 to yield the most coherent training signal, as well as how domain knowledge can be best generalized from this signal. Perhaps most interesting is the question of what sort of knowledge has actually been learned. Have the MRPN's learned some sort of adaptive mutational "cooling" schedule, specified by the distance the FRN vectors have deviated from their initialized state? How general is the learned control knowledge? How useful is it in guiding search for effective face recognition in similar problem classes? These questions are among those we are addressing in ongoing research.

At the same time, these preliminary results demonstrate that it is indeed feasible to automatically adapt evolutionary search to improve performance in a particular domain. Moreover, adaptation to the domain is accomplished via analysis of a training signal produced as a byproduct of evolution itself, suggesting that this entire process has the *potential* to be carried out by evolution directly, without the need for explicit "phases" (such as employed above). These findings indicate a promising direction for artificial evolution in complex domains such as image understanding.

References

[Baluja, 1995] Shumeet Baluja. Artificial neural network evolution: Learning to steer a land vehicle. In Lance Chambers, editor, *Practical Handbook of Genetic Algorithms: New Frontiers*, volume II, chapter 1. CRC Press, Inc., 1995.

[Fogel *et al.*, 1966] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence through Simulated Evolution.* John Wiley & Sons, New York, 1966.

[Fogel *et al.*, 1991] David B. Fogel, Lawrence J. Fogel, and Wirt Atmar. Meta-evolutionary programming. In R. R. Chen, editor, *Proceedings of the 25th Asilomar Conference on Signals, Systems & Computers*, volume 1, Washington, D.C., 1991. IEEE Computer Society Press.

[Fogel, 1995] D.B. Fogel. *Evolutionary Computation: Toward a New Philosphy of Machine Intelligence.* IEEE Press, Piscataway, NJ, 1995.

[Glickman and Sycara, 1996] Matthew R. Glickman and Katia P. Sycara. Self-adaptation of mutation rates and dynamic fitness. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, volume 2, page 1389. AAAI Press/MIT Press, 1996.

[Goldberg, 1989] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass., 1989.

[Holland, 1975] John H. Holland. *Evolution in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[Porto *et al.*, 1995] Vincent W. Porto, David B. Fogel, and Lawrence J. Fogel. Alternative neural network training methods. *IEEE Expert*, 10(3):16–22, 1995.

[Rechenberg, 1973] Ingo Rechenberg. *Evolutionstrategie: Optimeierung Technischer System nach Prinzpien der Biologischen Evolution.* Frommann-Holzberg Verlag, Stuttgart, 1973.

[Schwefel, 1995] Hans Paul Schwefel. *Evolution and Optimum Seeking.* John Wiley, Chichester, UK, 1995.

[Teller and Veloso, 1995] A. Teller and M. Veloso. Program evolution for data mining. In Sushil Louis, editor, *The International Journal of Expert Systems. Third Quarter. Special Issue on Genetic Algorithms and Knowledge Bases.*, pages 216–236. JAI Press, 1995.

Visual Learning for Landmark Recognition

Yutaka Takeuchi, Patrick Gros,
Martial Hebert, Katsushi Ikeuchi
School of Computer Science, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
hebert@cs.cmu.edu, ki@cs.cmu.edu
http://www.cs.cmu.edu/~{hebert,ki}

Abstract[1]

Recognizing landmark is a critical task for mobile robots. Landmarks are used for robot positioning, and for building maps of unknown environments. In this context, the traditional recognition techniques based on strong geometric models cannot be used. Rather, models of landmarks must be built from observations using image-based visual learning techniques. Beyond its application to mobile robot navigation, this approach addresses the more general problem of identifying groups of images with common attributes in sequences of images. We show that, with the appropriate domain constraints and image descriptions, this can be done using efficient algorithms as follows: Starting with a "training" sequence of images, we identify groups of images corresponding to distinctive landmarks. Each group is described by a set of feature distributions. At run-time, the observed images are compared with the sets of models in order to recognize the landmarks in the input stream.

1. Introduction

Recognizing landmarks in sequences of images is a challenging problem for a number of reasons. First of all, the appearance of any given landmark varies substantially from on observation to the next. In addition to variation due to different aspects, illumination change, external clutter, and changing geometry of the imaging devices are other factors affecting the variability of the observed landmarks. Finally, it is typically difficult to use accurate 3D information in landmark recognition applications. For those reasons, it is not possible to use many of the object recognition techniques based on strong geometric models.

The alternative is to use image-based techniques in which landmarks are represented by collection of images which are supposed to capture the "typical" appearance of the object. The information most relevant to recognition is extracted from the collection of raw images and used as the model for recognition. This process is often referred to as "visual learning".

Progress has been made recently in developing such approaches. For example, in object modeling [Gros et al.], 2D or 3D model of objects are built for recognition applications. An object model is built by extracting features from a collection of observations. The most significant features are extracted for the entire set and are used in the model representation. Extension to generic object recognition were presented recently [Carlsson, 1996].

Other recent approaches use the images directly to extract a small set of characteristic images of the objects which are compared with observed views at recognition time. For example, the eigen-images techniques are based on this idea.

Those approaches are typically used for building models of single object observed in isolation. In the case of landmark recognition for navigation,

[1] This research was sponsored in part by Office of Naval Research (ONR) under Contract N00014-95-1-0591 and, in part, by DARPA under contract DAAE07-96-C-X075 monitored by TACOM. Patrick Gros was supported by a NATO fellowship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR, DARPA or the U.S. Government.

there is no practical way to isolate the object in order to build models. Worse, it is often not known in advance which of the objects observed in the environment would constitute good landmarks.

A similar problem, although in a different context, is encountered in image indexing, where the main problem is to store and organize images to facilitate their retrieval [Lamiroy et al., 1996] [Schmid et al., 1996]. The emphasis in this case is the kind of features used and the type of requests that can be made by the user.

Our approach tries to combine these two categories of systems. In a training stage, the system is given a set of images in sequence. The aim of the training is to organize these images into groups based on similarity of feature distributions between images. The size of the groups obtained may be defined by the user, or by the system itself. In the latter case, the system tries to finds the most relevant groups, taking the global distribution of the images into account. In a second step, the system is given new images, which it tries to classify as one of the learned groups, or in the category of unrecognized images.

The basic representation is based on distributions of different feature characteristics. All these different kinds of histograms are computed for the whole image and for a set of sub-images. Tests are used to compare these histograms and define a distance between images. This distance is then used to cluster the images into groups. Each group is then characterized by a set of feature histograms. When new images are given to the system, the algorithm evaluates a distance between these images and the groups. The system determines to which group this image is the closest, and a set of thresholds is used to decide if the image belongs to this group.

The main goal of the work presented here was to explore the use of tools and methods in the field of image retrieval to the problem of landmark recognition. It is clear that the global architecture of the system is close to that of object recognition systems [Gros et al.]: a training stage in which 3D shape, 2D aspects, or groups, are characterized is followed by a recognition stage in which this information is used to recognize the models, objects or groups in new images. The difference comes from the wide diversity of the images and from the groups which are not reduced to a single aspect of an object. The two challenging tasks which we concentrate on in the remainder of the paper are to define these groups more precisely as sets of images, and to build automatically a characterization for each group.

The first section of the paper deals with the feature distributions, their computation and their comparison. The second section addresses the computation and the characterization of groups. The third section concerns the classification of new images according to the groups previously defined. Experimental data and results are presented in the fourth section.

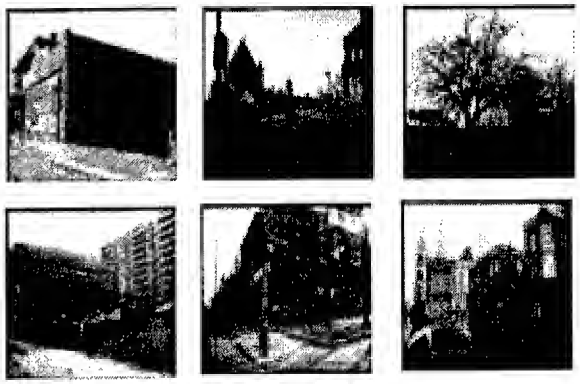


Figure 1: Sample set of images from a typical training sequence. The complete training sequence contains 176 images taken over approximately 800 meters.

2. Representing Images

In this section, we give a brief overview of the features used for representing images. Since an individual feature can be characteristic of an aspect of an object, but probably fails to characterize well a set of aspects, we use a statistical description of a large number of features as our basic representation Representing Feature Distributions

Five basic features are currently used: distributions of normalized color and intensity, edges, segments, and parallel segments. Additional features can be added as needed. The basic image representation is a set of feature distributions. Edges are computed using Deriche's edge detector [Deriche, 1987]. Segments are computed as a polygonal approximation of the edges [Horaud et al., 1990].

Among the characteristics computed from these five features are: color and grey levels; edge density, orientation, length and position; segment density, orientation, length, and position; parallel segment density, orientation, length, and position, and, finally, the angles between adjacent segments.

Feature densities are computed in two ways: first as the ratio of the number of features (edges, segments and parallel segments, respectively) to the area of the image or sub-image concerned; second as the ratio of the sum of all the feature lengths to the area of the image or sub-image. All the other measurements are computed and the results stored in histograms. Each bucket of the length histograms indicates how many features a length has in a given interval.

The position of a particular object in an image may vary substantially between observations. Therefore, it is important to build the representation in a way that allows for different placements of the object with similar resulting feature distributions. This is done by subdividing the image into smaller chunks, in which the feature distributions are computed. All these histograms and densities, except those relative to feature position, are computed for the whole image and for the sub-images obtained by dividing the image by 4, then 9, and then 16. The position histograms are computed only for the global image, i.e., 90 densities and 333 histograms are computed to characterize each image.

2.1. Comparing Feature Distributions

The feature distributions from two images are compared using a distance similar to the chi-square distance. This distance, in its simplest form, evaluates the probability that two sets of data, here the histograms or the densities, are derived from the same theoretical distribution. If the distributions are $h = (h_i)$ and $l = (l_i)$, their difference is computed as [Press et al., 1992]: $\mathrm{d}(h, l) = \Sigma_i (h_i - l_i)^2 / (h_i + l_i)$.

The main problem is to derive a global distance between two images from the individual distances computed for each type of density and histogram. The distance d_{ij} between two images i and j is defined as the linear combination of the distances between individual feature distributions: $d_{ij} = \Sigma_k \lambda_k \mathrm{d}(h, l)$, where the sum is taken over all the feature distributions used for representing the images.

When nothing is known about the distributions and their range of variation, all the weights can be taken equal to one. This simple approach gives good results in practice, but a better approach is to compute the weights based on the relative scales of the feature distributions. For each kind of density or histogram, the distance between every pair of images is computed, and the variance $\sigma^2{}_k$ of these distances is derived. The coefficient relative to this particular distribution can be chosen as $\lambda_k = 1/\sigma^2{}_k$. This choice of weights has the effect of normalizing the distributions and of assigning the same relative importance to all the partial distances used.

3. From Images to Landmarks

Given a sequence of images, we now want distinctive landmarks, that is, we want to split the sequence into groups of images, and find a characterization of each of these groups which allows further classification.

This step is difficult to do fully automatically in general. The main reason is that there is not a task-independent definition of the type of image groups that are needed. Our approach is to use task constraints to guide the grouping process. Specifically, given an initial grouping of images, we select groups based on three constraints. First, only the groups that contain a large enough number of images from different aspects are retained. Second, groups that do not provide significant discrimination are discarded. This is important to ensure that, at recognition time, only the groups that can be easily distinguished are used as models. Finally, the recorded sensor position for each training image is used for ensuring that the groups are spatially coherent.

3.1. Computing Initial Image Groups

Once the distance matrix is computed, a simple agglomerative method is used to split the image set into initial groups. First each image is put in a different group. Then the two closest groups are grouped and the distance matrix is updated. Finally, the algorithm iterates the previous step until an ending condition is verified.

Let |L| denote the number of elements of the image group L. The update of the matrix consists of suppressing the two lines and two columns i and j corresponding to the groups I and J which are grouped, and then adding a new line and column for the new group formed. The diagonal term of the new line added is:

$$\frac{|I|(|I|-1)d_{ii}+|J|(|J|-1)d_{jj}+2|I||J|d_{ij}}{|I|(|I|-1)+|J|(|J|-1)+2|I||J|} \quad (1)$$

The non diagonal term corresponding to the new group and a group k is:

$$\frac{|I|}{|I\cup J|}d_{ik}+\frac{|J|}{|I\cup J|}d_{jk} \quad (2)$$

These formulas show that, at each iteration, the only information needed is the distance matrix and the number of elements in each group. Each term of the matrix is thus the mean distance between the images of two groups, or the mean distance of the images of a same group.

3.2. Controlling the Grouping Algorithm

The grouping algorithm described above is general. In particular, it does not incorporate a control structure that stops the grouping process when groups of images corresponding to recognizable landmark are formed. An automatic method was developed for controlling image grouping.

Given a set of image groups, let us denote the distance matrix by (d_{ij}) and the number of images of the group i by n_i. If the images used to learn the groups form a representative sample, and if they are spread nearly uniformly in representation space, the probability that an unknown image will be classified in the group i (p_i) or in no group at all (p_0) can be evaluated. If n denotes the number of groups:

$$p_i=\frac{d_{ii}}{\sum_j d_{jj}+\frac{2}{n-1}\sum_{j\neq k}d_{jk}} \qquad p_0=\frac{\frac{2}{n-1}\sum_{j\neq k}d_{jk}}{\sum_j d_{jj}+\frac{2}{n-1}\sum_{j\neq k}d_{jk}} \quad (3)$$

These formulas state that the probability p_i that a new image belongs to a group is proportional to the size d_{ii} of this group, and that the probability p_0 of being in no group is proportional to the distances d_{jk} between the groups. The factor $2/(n-1)$ is used to compensate the number of non-diagonal terms of the distance matrix with respect to the number of diagonal terms.

In the case where the images are known not to have a uniform distribution in a region of their representation space, this can be taken into account by using these other formulas:

$$p_i=\frac{n_i d_{ii}}{\sum_j (n_j+1)d_{jj}+\frac{2}{n-1}\sum_{j\neq k}d_{jk}}$$
$$p_0=\frac{\frac{2}{n-1}\sum_{j\neq k}d_{jk}}{\sum_j (n_j+1)d_{jj}+\frac{2}{n-1}\sum_{j\neq k}d_{jk}} \quad (4)$$

In these new formulas, not only the size of the groups is taken into account but also their density, which is proportional to n_i. The probability p_0 is also a function of the size of the groups: for the same distances between the groups, the smaller the groups, the bigger their density, and the smaller is the probability of having a new image between them.

The evaluation function is the entropy associated with this set of probabilities $S = -\Sigma_i p_i \ln p_i$, and the process is stopped when this entropy is maximal. This maximizes the information provided to the user by each classification request.

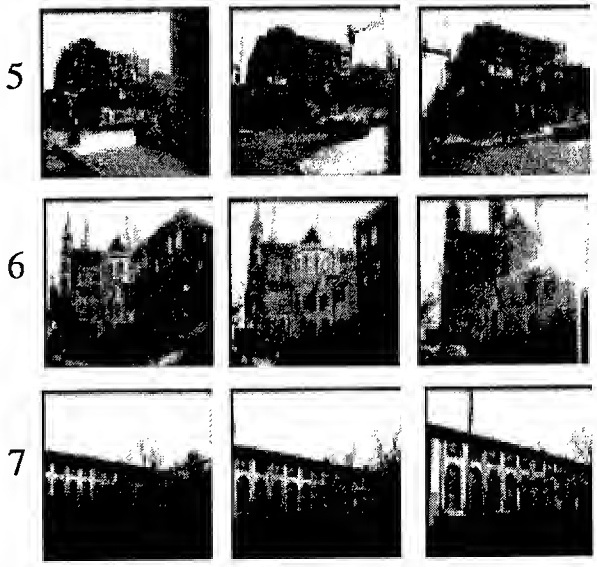


Figure 2: Three of the groups extracted from the training sequence of Figure 1. Only three images are shown for each group.

3.3. Using Domain Constraints

An important constraint in the context of landmark recognition is that the images are ordered in the training sequence. In fact, the position of the vehicle is recorded for each image. Using this information, grouping is limited to images for which the vehicle positions were close to each other, thus ensuring spatial consistency.

In general, given an image, it would be necessary to consider other images for grouping in a radius around the corresponding vehicle position. In practice, images in the training sequence are digitized at approximately equal intervals. As a result, it sufficient to consider for grouping with image i only images j such that $|i - j| < r$, where r is the maximum extent of observability of any given landmark. This constraint reduces the computational complexity of the grouping algorithm, and it guarantees that the image groups correspond to spatially coherent landmarks.

4. Recognition

Given a set of image groups C_i, characterized by their mean vector v_i, their eigenvalues $\lambda_{i1}...\lambda_{il}$, and their eigenvectors $v_i^1 ... v_i^l$, the problem is to compare these groups to a new image whose characteristic vector is v. The eigenvalues are positive, and the eigenvectors of a group are a family of orthonormal vectors, and $v - v_i$ may be decomposed with respect to this family: $v - v_i = \Sigma_j a_j v_i^j + r$. The distance between v and the group C_i is derived as:

$$d(C_i, v) = \sqrt{\sum_j \frac{a_j^2}{\lambda_i^j + 1} + \|r\|^2} \qquad (5)$$

This formula allows us to find which is the closest group to an image. The problem is then to decide if the image really belongs to this group. We again use the task constraints. Intuitively, consecutive images should be classified in a consistent manner. Since we know the spatial extent from which a particular group of images is visible in the training sequence, we can eliminate the inconsistent classification results as unreliable.

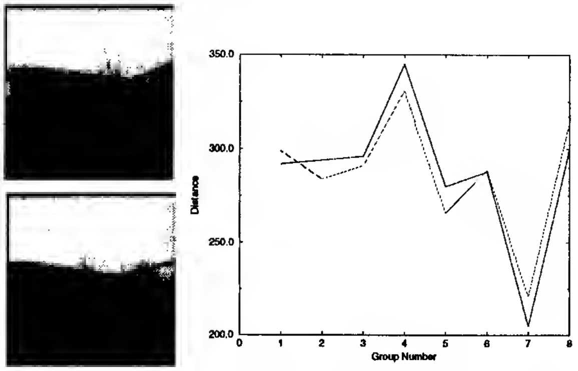


Figure 3: Comparison of two new images (top) with the groups shown in Figure 2. The distance graph of the left image is shown as a solid line. The images are correctly classified in group 7.

5. Examples

The recognition algorithm was tested using several sequences taken from a moving vehicle. Figure 4 shows sample images from a test sequence taken over the same course as the training sequence of Figure 1. A total of 68 images were digitized from the test sequence at regular intervals. The test set of images was segmented manually into subsets corresponding to the landmarks identified in the training sequence. This provided the “ground truth” for evaluating the performance of the algorithm. Vehicle position was recorded using the INS system onboard a HMMWV used in a separate Unmanned Ground Vehicle (UGV) project [Hebert et al., 1996].

Using the algorithm outlined above, all the images in the test sequence were compared with the landmarks found in the training sequence. The images corresponding to a distinct distance minimum are labeled with the corresponding landmark number. Images that do not correspond to a distance minimum for any landmark are left unclassified. As we noted earlier, our goal is not to label every image but rather to correctly recognize the ones corresponding to the most salient landmarks.

Figure 5 illustrates the classification algorithm for three different landmarks. The graphs show that, for those three landmarks, the distance minimum is attained at the correct images in the test sequence. Figure 7 shows a view of all the landmarks recognized in the path travelled in the test sequence.

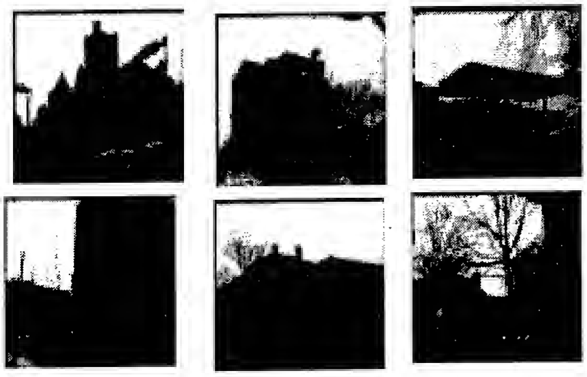


Figure 4: Sample images from a test sequence. The complete sequence is 68 images over a course similar to the one used in Figure 1 for the training sequence. The orientation of the camera, and the illumination characteristics are substantially different from those of the training sequence.

Four landmarks are recognized in this example. Figure 7 also illustrates the potential use of landmark recognition for position estimation.

Two types of errors may occur during recognition. First, images that should match a landmark are matched to a different landmark. We call those images misclassified images. Second, images that should match a given landmark are not matched with any landmark. We call this second class of images unclassified images. To reduce the error rate, we use the sequential constraint described earlier. This constraint is quite effective in practice. Figure 6 shows the error statistics for the recognition algorithm with and without sequential constraint.

All the examples given so far involve images taken in urban environments. We have also conducted experiments in natural environments by collecting training sequences, extracting groups of distinctive images, and recognizing them in test sequences. Figure 8 shows two example groups computed from a typical training sequence. The error rate, is comparable to the one obtained in urban scenarios.

6. Conclusion

Results on image sequences in real environment show that visual learning techniques can be used for building image-based models suitable for recognition of landmarks in complex scenes. The

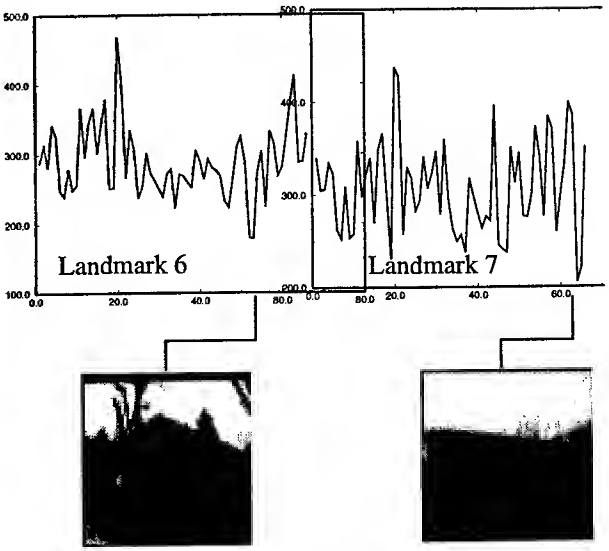


Figure 5: The classification algorithm illustrated on two landmarks. Each graph shows the distance between all the images of the test sequence of Figure 4 the groups found in the training sequence (Figure 2.) The graphs are shown for landmarks 2 and 7. The graphs show that the distance is minimum for the correct landmark.

Urban Environment:

	distance only	with sequential constraint
correct	72%	93%
misclassified	19%	0%
unclassified	9%	7%

Natural Environment:

	distance only	with sequential constraint
correct	84%	97%
misclassified	6%	0%
unclassified	10%	3%

Figure 6: Performance of the recognition algorithm on the two example sequences. Images are labeled as "misclassified" if they are matched to the wrong group; they are labeled as unclassified if they belong to a group but are not matched.

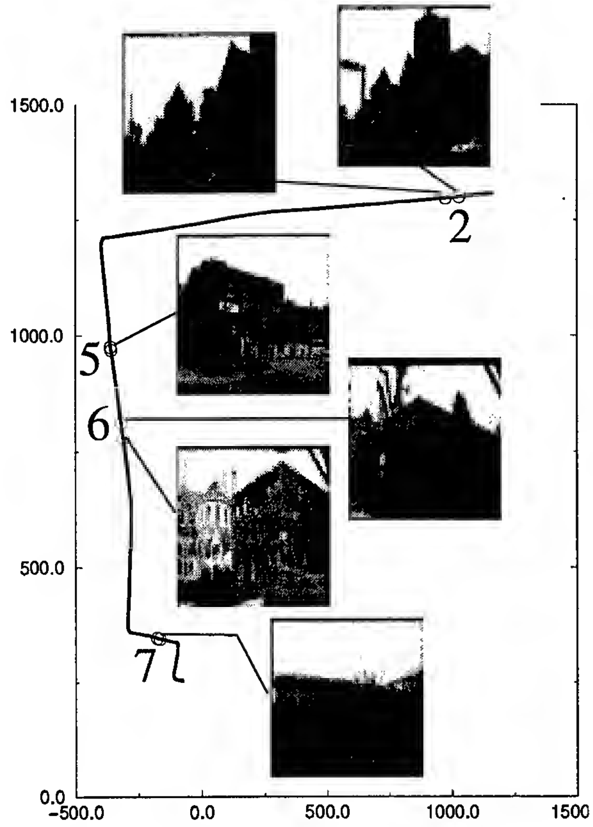


Figure 7: Overhead view of the path followed while collecting the images of the test sequence of Figure 4 (distances are indicated in meters.) Four landmarks are correctly identified. Example images from the test sequence are shown for each landmark.

Figure 8: Images of one of the landmarks found in a sequence of images in a natural environment.

approach performs well, even in the presence of significant photometric and geometric variations, provided that the appropriate domain constraints are used. In the case of mobile robot navigation, domain constraints include the sequential nature of the images, and the discriminability of landmarks.

Our goal is to demonstrate the use of landmark recognition for navigation. Specifically, we will show that rough position estimation and navigation based on the relative positions of landmarks can be achieved using image-based landmark recognition. Several limitations of the approach need to be addressed. First of all, rejection of unreliable groups need to be improved. In particular, the selection of the parameters controlling the grouping need to be implemented in a principled manner. Second, images that do not contribute information should be filtered out of the training sequences.

References

[Carlsson, 1996] S. Carlsson. Combinatorial Geometry for Shape Representation and Indexing. *Proc. International Workshop on Object Representation for ComputerVision*. Cambridge, England, April 1996.

[Deriche, 1987] R. Deriche *Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector.* Int. Journal of Computer Vision 1(2), pages 167--187, 1987

[Gros et al.] P. Gros, O. Bournez and E. Boyer. *Using Local Planar Geometric Invariants to Match and Model Images of Line Segments.* To appear in Int. J. of Computer Vision and Image Understanding.

[Hebert et al., 1996] M. Hebert, C. Thorpe, A. Stentz. *Intelligent Unmanned Ground Vehicle.s.* Kluwer Publishers. 1996.

[Horaud et al., 1990] R. Horaud, T. Skordas and F. Veillon. Finding Geometric and Relational Structures in An Image *Proc. of the 4th European Conf. on Computer Vision,* Antibes, France April 1990

[Lamiroy et al., 1996] B.Lamiroy and P.Gros. Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing. *Proc. of the 4th European Conf. on Computer Vision*, Cambridge, England, pages 59--70, vol. 1, April 1996.

[Press et al., 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recipes in C.* Second Edition, Cambridge University Press 1992.

[Schmid et al., 1996] C. Schmid and R. Mohr. Combining Greyvalue Invariants with Local Constraints for Object Recognition. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA. pages 872--877, June 1996

Evolutionary Learning for Orchestration of a Signal-to-Symbol Mapper

Astro Teller* and **Manuela Veloso**
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

A wide variety of machine learning mechanisms create multiple models that must be reconciled, chosen among, or in some cases, *orchestrated.* In its most general form, this orchestration problem can be seen as part of the multi-agent learning problem. This paper introduces particular complexities of automatically sub-dividing a problem, developing multiple solutions to each sub-problem, and then orchestrating the sub-solutions into a complete solution. This paper establishes, through a series of experiments, that this divide and conquer strategy can be done automatically, the effectiveness of four introduced, specific techniques for learning orchestration, and that the orchestration of sub-solutions is a rich, intriguing area for machine learning study.

1 Introduction

There are many cases in which a task to be approached with machine learning techniques can be or must be solved in more that one "piece." Learning a team of robotic soccer players is a good example of a task that could conceivably be done as a single agent, but lends itself very naturally toward learning sub-solutions and *then* (or in addition) learning to ensure the mutual suitability of these sub-solutions. This insurance of mutual suitability is the *orchestration problem.*

This paper will focus on PADO, a evolutionary computation framework designed specifically for signal classification (e.g., [Teller and Veloso, 1997, Teller and Veloso, 1995b, Teller and Veloso, 1995a]). As a process of divide and conquer, PADO evolves multiple pools of sub-solutions and then orchestrates one or more learned models from each pool. The question we investigate in this work is, "What opportunities are there for learning in the orchestration process and how much improvement can this learning provide?" Our experiments demonstrate that orchestration is an important issue and that learned orchestration can dramatically improve generalization performance.

2 PADO

PADO, Parallel Algorithm Discovery and Orchestration, is an evolutionary learning paradigm specifically designed for signal classification. At the highest level of description, PADO has two main learning components: algorithm discovery, and orchestration. PADO does the algorithm discovery through a process of program evolution as pictured in Figure 1.

PADO evolves programs in a PADO-specific graph structured language. At the beginning of a learning session, the main population is

*This research was sponsored partly by the Office of Naval Research (ONR) under Contract N00014-95-1-0591 and partly by the Fannie and John Hertz Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR, the Fannie and John Hertz Foundation, or the U.S. Government.

filled with P programs that have been randomly generated using a grammar for the legal syntax of the language. All programs in this language are constrained by the syntax to return a number that is interpreted as a confidence value. The exact structure of the language, *Neural Programming*, and the associated recombination process is detailed in [Teller and Veloso, 1996]. For the purposes of this paper it suffices to understand that the PADO architecture has the ability to *learn* a number of programs that do well at solving part or all of a particular problem.

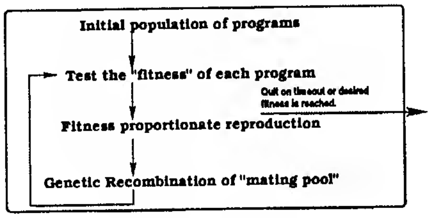

Initial population of programs
Test the "fitness" of each program
Quit on timeout or desired fitness is reached.
Fitness proportionate reproduction
Genetic Recombination of "mating pool"

Figure 1: The evolution of programs

The goal of the PADO architecture is to learn to take signals as input and output correct class labels (i.e. perform the signal-to-symbol mapping). *When there are C classes to choose from, PADO starts by learning C different "pools" of* discrimination programs. System$_I$ is responsible for taking a signal as input and returning a confidence that class I is the correct label. System$_I$ is built out of one or more programs for pool I, learned by PADO. Each of these programs does exactly what the system as a whole does: it takes a signal as input and returns a confidence that label I is the correct label (rather than returning a value between 1 and C). The reason for this seeming redundancy can be found in [Teller and Veloso, 1997]. PADO performs signal classification by orchestrating the responses of the C systems.

The basic justification for subdividing work as PADO does is: it is usually preferable to search C spaces of size 2^K rather than one space of size 2^{CK} $(K > 1)$. In classification problems it is possible to automatically divide up a problem: one classification problem of C classes can always be decomposed into C discrimination problems. A program in System$_I$ learns to say "high confidence" when shown an instance of signal class I and "low confidence" when presented with a signal from any other class.

A simple orchestration scheme that was used in earlier reported PADO work (e.g., [Teller and Veloso, 1997]) is depicted in Figure 2. System$_I$ is built from the S programs that best (based on the training results) learned to recognize the instances of class I. The S responses that the S programs return on seeing a particular signal are combined into a weighted average. This average is interpreted as the confidence of system$_I$ that the signal in question is from class I. The responses of the C pools are weighted and combined in a similar way.

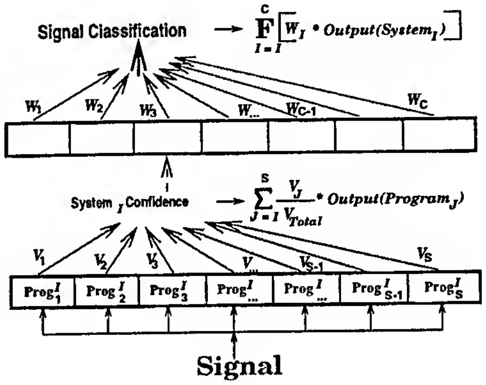


Figure 2: PADO's old orchestration strategy

In this paper we will concentrate on the higher level orchestration in PADO. We will make the simplifying assumption that exactly one program will be chosen from each discrimination pool and orchestrated one of the ways outlined in the next section. This simplification of PADO's orchestration for explanation purposes is shown in Figure 3. The following section will address issues of which programs to use, how to use them, and how to bias them through evolution itself.

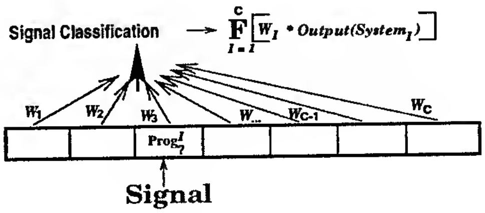

Signal Classification
$\xrightarrow{} \mathop{\mathbf{F}}_{I=1}^{C}\left[W_I * Output(System_I)\right]$
W_1
W_2
W_3
$W_{...}$
W_{C-1}
W_C
$Prog_?^I$
Signal

Figure 3: Part of PADO's new orchestration strategy

3 Orchestration Options

The orchestration options that we investigate in the course of this work include: 1) default orchestration, 2) evolved orchestration, 3) learned weight orchestration, 4) learned program orchestration, and 5) combination strategies.

3.1 Default Orchestration

In the simplest option, *default orchestration*, PADO picks the best program from each discrimination pool and orchestrates them using a fixed function with fixed parameters (i.e. no learning). This orchestration will be the baseline against which we will measure other techniques. The first step in default orchestration is to pick the best program from each discrimination pool to be in the orchestrated set. Specifically, the procedure is to sort the programs in each discrimination pool according to training set fitness and choose the top ranked program from each discrimination pool. This program set is orchestrated with a simple function using reasonable, fixed coefficients, hereafter referred to as weights.

Each program X has a fitness, F_X, (averaged over all training examples) that ranges between 0.0 and 1.0. Let us give each program X selected as part of the final PADO classifier an orchestration weight of $W_X = F_X$. For a particular test signal, each program is shown the signal and returns a response R_X between R_{min} and R_{max}. Let us refine R_X according to:

$$R_X := (W_X * R_X) - (1 - W_X) * \frac{R_{max} - R_{min}}{2} \tag{1}$$

Now if the program from class i has the highest R_X, PADO concludes the test signal is from class i. However, because of equation 1, PADO "listens more attentively" to programs that are, on average, "more reliable" (i.e. have higher fitness). Notice that this default orchestration makes no attempt to pick: the best weights for each program, the best programs to orchestrate, or a fitness function that promotes orchestration.

3.2 Evolved Orchestration

An alternative to learning how best to orchestrate a number of programs, or which programs to orchestrate, is to try to change the basic learning of the programs so that the programs that perform best on the training set will also be the best orchestrated exactly as they are. Because PADO learns programs in an evolutionary framework, this amounts to incorporating the demands of a particular orchestration strategy into the *fitness function.* At each generation, after computing the fitness of each evolving program, PADO can assign each program a new fitness, based on its ability to orchestrate with random individuals from the other discrimination groups.

For each discrimination pool i, PADO creates K biased-random[1] groups (using one program from each of the other $C - 1$ discrimination pools) called $P_1^i..P_K^i$. For each discrimination group i, for each program X in the group:

$$F_X^{new} = \frac{\sum_{j=1}^{K} Eval(X \cup P_j^i, \text{Weights})}{K} \tag{2}$$

Eval($X \cup P_j^i$, Weights) is the percentage of training examples the orchestrated PADO program set of $\{X \cup P_j^i\}$ correctly classifies, orchestrated with Weights (just the F_X's in this case). This value is an approximation to how well X "orchestrates in general", relative to other programs in the same discrimination pool.

From this point on, PADO follows the default orchestration strategy except that the program from each discrimination pool chosen for orchestration will be the *best* based on F_X^{new} rather than old fitness. In evolved orchestration, F_X^{new} replaces F_X in the entire process.

3.3 Learned Weight Orchestration

In *learned weight orchestration* PADO tries to find the best set of weights for the chosen set of programs to be orchestrated. We get one degree of freedom for each program to be orchestrated

[1] Picked randomly, biased for higher fitness individuals.

by allowing each W_X to vary between 0 and 1 (instead simply setting it to F_X).

To begin, this strategy will set the *Progs* to orchestrate to be the best program from each discrimination pool based on training set fitness. Next, the strategy will initialize the *Weights* and *BestWeights* for orchestration to $W_X = F_X$ for $W_1..W_C$ just as in the default orchestration case. Now we can search for better values of $W_1..W_C$. Now repeat S times:

```
1. Pick $i$ between 1 and C.
2. Change Weights[$i$]
3. If Eval(Progs,Weights) >
     Eval(Progs,BestWeights),
   then update BestWeights[$i$]
```

Do step 1 based on previous successes changing that element of the vector. Change step 2 in the direction that most recently helped when changing this vector element. Anneal step 3 so that local minima are partly avoided.

3.4 Learned Program Orchestration

In *learned program orchestration* weights are fixed at their default values ($W_X = F_X$) and PADO tries to find the set of programs that best fit those weights. We get one degree of freedom per class (per orchestration weight) by allowing the program to orchestrate for discrimination class i to vary over all of the programs in discrimination pool i.

To begin, this strategy will initialize the *Progs* and *BestProgs* to be the best program from each discrimination pool based on standard F_X. The orchestration weights (*Weights*) will be set for each program X to be orchestrated to its F_X. Now we can search for better programs X_1 .. X_C. Now repeat S times:

```
1. Pick $i$ between 1 and C.
2. Exchange Progs[$i$][2] for another
     program from the discrimination
     class $i$ pool.
3. If Eval(Progs,Weights) >
     Eval(BestProgs,Weights),
   then update BestProgs[$i$]
```

Do step 1 based on previous successes changing that element of the vector. Pick step 2 with a bias toward more highly fit programs. Anneal step 3 so that local minima are partly avoided

3.5 Example Combination Strategies

It is possible to combine variations like the ones described above in a variety of ways. For example, we can first search through program space with a set of fixed weights (section 3.4), and then once having found this cooperative set of programs we can then search through weight space (section 3.3) for this discovered set of programs for the optimal weight set for those programs. This combination strategy will be referred to in the following experiments section as simply "Learned Prog..Weight Orchestration." Another combination possibility is to evolve better orchestration (section 3.2) and then, at orchestration time, to do a search to find the best program set (section 3.4).

4 Experiments

All of the following experiments were performed with the following fixed parameters: population size 600, crossover 48%, mutation 48%, 12000 program combinations considered when learning program orchestration, 12000 weight combinations considered when learning weight orchestration, and no drift between the subpopulations. Every curve in every figure in this section is an average over at least 50 separate, independent runs. Programs were exposed to up to 200 randomly generated training signals[3]. The test set consisted of a different set of 500 randomly generated signals.

4.1 Four Classes

In this problem, a simple wave form type was used as the input. The location of the two spikes in the otherwise highly damped wave move about evenly over the signal. The signal has four possible classes that are encoded as a two bit binary number in the two wave spikes, low amplitude means "0", high variance means

[3] See [Teller and Andre, 1997] for details.

"1." Figure 4 shows one randomly selected example from each of these four classes.

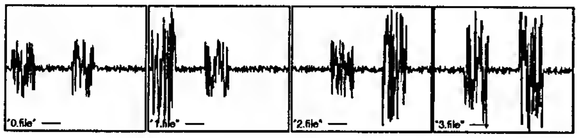


Figure 4: One example from each of the four classes.

Figure 5 has a number of different curves. This is a graph measuring the computational effort required to achieve a given level of task performance. For example, the "default orchestration" curve has a point at (0.7,15). This is to be read as "In order to achieve a 70% generalization test set performance using the default orchestration strategy, PADO must run for, on average, 15 generations."

The "default orchestration" curve is the one against which the other curves should be measured and shows PADO's performance when no orchestration search is done. Figure 5 also shows four learning strategies: learned program orchestration, learned weight orchestration, evolved orchestration, and evolved..learned program orchestration (do evolved orchestration and *then* do learned program orchestration).

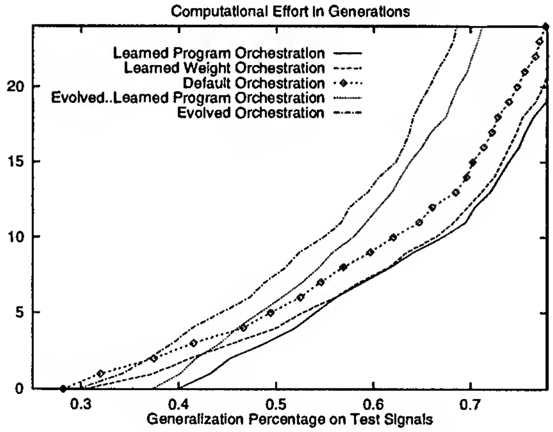


Figure 5: Four Class Results.

The most important feature of Figure 5 is that the evolved orchestration actually does *worse* than the default orchestration strategy (i.e. no learning). While PADO does better in the "evolved..learned program orchestration" paradigm than the "evolved orchestration" paradigm, this is still worse than simple default orchestration.

Though this paper will not try to substantiate this claim, here is an intuition for why putting the orchestration directly into the fitness function actually *hurt* PADO's performance. In the default orchestration case there was some orchestration going on, just no orchestration *learning*. In addition, PADO begins by subdividing its population, and thereby the problem, into several easier sub-problems. This subproblem division is reasonable for classification and PADO enforces it throughout the entire run. When fitness gets tied to orchestration instead of discrimination, *PADO loses exactly those distinctions.* So we argue that PADO may have gained something through evolved orchestration, but *at the cost of losing the whole mechanism of divide and conquer* that made orchestration important in the first place.

4.2 Eight Classes

Our next experiment will take a second look at the "Learned Program Orchestration" and "Learned Weight Orchestration" strategies. In this second experiment we preserve all the details of the previous experiment except that the problem has more classes and is a harder problem of the same type. The natural extension of the previous experiment along these constraints is to use an eight class signal classification problem with similar damped waves and spikes. There are now three spikes in each wave form, still in variable locations, and the three spike amplitudes encode the class (low amplitude is "0" and high amplitude is "1"). Figure 6 shows one randomly selected signal from each class.

Figure 7 shows the average performance results of four different strategies. Again, the benchmark performance is the curve labeled "Default Orchestration" The new strategy that this section introduces is labeled "Learned Prog..Weight Orchestration" and is simply a search through program space followed by a search through weight space.

Figure 7 has two main points of interest. The

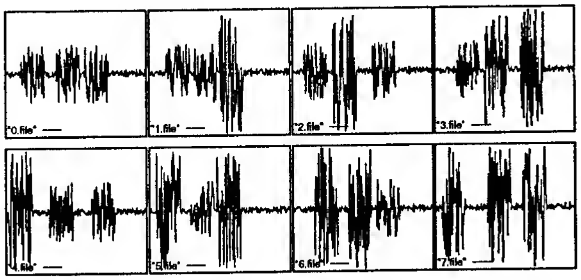


Figure 6: One example from each of the eight classes.

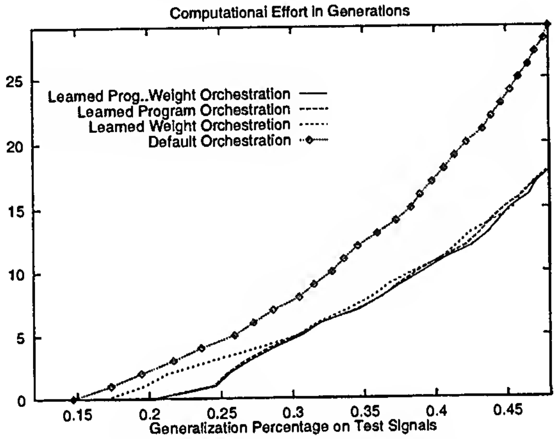


Figure 7: Eight Class Results.

first to notice is that the problem is indeed harder; generation 30 produces a lower generalization PADO system. The second point of interest is that, again, all three of the orchestration learning procedures improve considerably over default orchestration.

4.3 Ten Classes

As a further experiment let us now try a different kind of signal and increase the number of classes in the domain again. For this experiment we will test the same four orchestration strategies on a signal that expresses its class by the slope of its wave form. Since this is not directly observable by the PADO constituent programs and so must be inferred from multiple local observations, this is still not a trivial problem. Figure 8 shows one randomly selected example from each class.

Once again, we can see from the curves in Figure 9 that all three orchestration learning strategies do better than orchestration without learning. In this experiment the search in weight space was less effective than the search in program space, though still helpful. By the time we reach a PADO system generalization level of 60%, the standard orchestration method is a much steeper slope than the "Learned Program Orchestration" and "Learned Prog..Weight Orchestration" strategies.

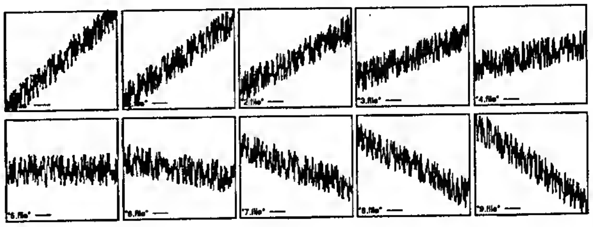


Figure 8: One example from each of the ten classes.

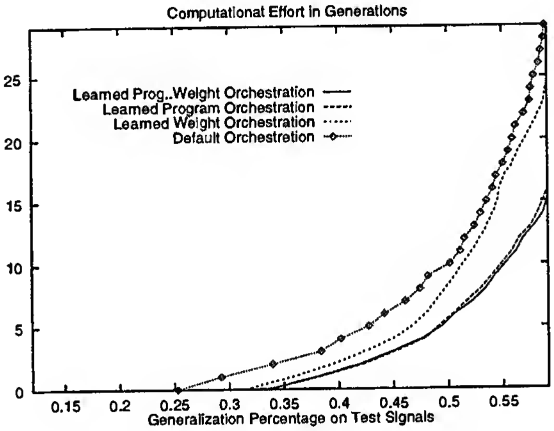


Figure 9: Ten Class Results.

In the last two experiments discussed in this paper, the "Learned Prog..Weight Orchestration" curve is within statistical noise of the "Learned Programs Orchestration" curve for every level of generalization. The explanation of this is that because in the "Learned Prog..Weight Orchestration" strategy the best program set is found *for a particular fixed weight vector* and then the weights tuned for that program, it is not surprising that for the program set selected, the best weight vector is almost always the one under which it was optimized.

5 Related Work

Examples of the evolution of behavior coordination can be found in [Haynes *et al.*, 1995, Haynes and Sen, 1996]. In these examples, the "teams" are explicitly grouped, leading to a natural incorporation of cooperation into the fitness function. [Wolpert, 1992] gives a very thorough theoretical account of "stacked generalization." The general conceit of stacked generalization is that instead of having a learning algorithm entirely solve a problem, one or more models can be used to partially solve the problem. Then, the output of that model(s) can be "stacked" as inputs to a new learner. Though the description is very different, the orchestration problem can be seen as a specific difficulty in stacked generalization. This work has attempted to address some of these specific difficulties.

6 Conclusions

We addressed the *orchestration problem* in the context of PADO, a divide and conquer evolutionary technique for signal classification. In general, the issues studied apply to devide-and-conquer learning problems in which putting the sub-solutions together again (i.e., orchestrating them) is non-trivial.

Three experiments on distinct signals demonstrated the feasibility of PADO's divide and conquer strategy. The failure of the evolved orchestration procedure suggested PADO's preferability to unconstrained learning. These experiments provided a specific justification for maintaining a population: orchestration puts the options a population provides to good use. This work introduced four specific techniques for orchestration learning which demonstrated that orchestration is an important issue and that learned orchestration can provide dramatic generalization improvements.

References

[Haynes and Sen, 1996] Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predators and prey. In Gerhard Weißand Sandip Sen, editors, *WEISS96*, pages 113–126. Springer Verlag, Berlin, 1996.

[Haynes *et al.*, 1995] Thomas Haynes, Roger Wainwright, Sandip Sen, and Dale Schoenefeld. Strongly typed genetic programming in evolving cooperation strategies. In Stephanie Forrest, editor, *ICGA95*, pages 271–278, San Mateo, CA, July 1995. Morgan Kaufman.

[Koza, 1992] J. Koza. *Genetic Programming.* MIT Press, 1992.

[Teller and Andre, 1997] Astro Teller and David Andre. The rational allocation of trials. 1997. Submitted for review to the 1997 International Joint Conference on Artificial Intelligence.

[Teller and Veloso, 1995a] A. Teller and M. Veloso. Algorithm evolution for face recognition: What makes a picture difficult. In *Proceedings of the International Conference on Evolutionary Computation.* IEEE Press, 1995.

[Teller and Veloso, 1995b] A. Teller and M. Veloso. Program evolution for data mining. In Sushil Louis, editor, *The International Journal of Expert Systems. Third Quarter. Special Issue on Genetic Algorithms and Knowledge Bases.*, pages 216–236. JAI Press, 1995.

[Teller and Veloso, 1996] Astro Teller and Manuela Veloso. Neural programming and an internal reinforcement policy. In *First International Conference on Simulated Evolution and Learning*, pages 279–86. Springer-Verlag, 1996.

[Teller and Veloso, 1997] A. Teller and M. Veloso. PADO: A new learning architecture for object recognition. In K. Ikeuchi and M. Veloso, editors, *Symbolic Visual Learning.* Oxford University Press, 1997.

[Wolpert, 1992] D. H. Wolpert. Stacked generalization. In *Neural Networks*, volume 5, pages 241–259. Pergamon Press, 1992.

Multi–Spectral Imaging Filters

L.J. Denes, M. Gottlieb, B.Kaminsky, P. Metes, Z.K. Kun,
M. Capizzi, J. Hibner, D. Purta, A.M. Guzman
Carnegie Mellon Research Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213
email: ld2s@andrew.cmu.edu
http://www.cs.cmu.edu/afs/cs/usr/brajovic/www/lab/cmu-muri.html

Abstract[1]

We have assembled a multi-spectral imaging system operating in the visible to near IR range utilizing an existing acousto-optic tunable filter (AOTF). This configuration has been characterized, yielding design optimization information. Critical data include spatial and spectral resolution, out-of-band rejection, efficiency, field of view, and bandwidth. The design goal is efficient operation over nearly two octaves of wavelength, and superior image quality. Two major issues were successfully addressed: 1) the method of applying the multiple electrical RF control signals to the AOTF transducer to fully exploit the multispectral capability, and 2) object identification using color signature signal deliverted by the AOTF.

1. Objective

This program task incorporates the spectral (color) dimension into the object recognition process. A programmable optical filter is utilized at the system's front end to reduce the computational load and its resulting bottlenecks in future automated vision systems. By filtering the incoming scene according to its spectral composition, a large amount of undesirable background clutter can be removed prior to digital processing. Figure 1 is a schematic representation of the process. Enhanced performance is anticipated in a variety of applications, including human sensory augmentation systems for driver assistance. This recognition process will more closely copy the human observer in its ability to extract and track objects.

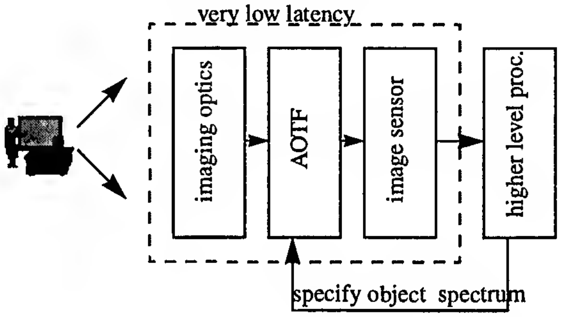


Figure 1: Object recognition using color discrimination.

2. Approach

Multi-spectral imaging is based on a "smart filter" concept that utilizes features available with the acousto-optic tunable filter (AOTF). Such smart filters provide dynamically programmable spectral image selection, which in combination with new computational sensors, can result in new strategies for object recognition and tracking. The key features of the AOTF which make them so effective for this purpose, include all-solid state construction and electronic operation, rapid random access of wavelength, simultaneous multiple-wavelength operation, and electronically adjustable pass bandwidth; these features are readily adaptable to computer control. In addition, the devices are compact and robust, and easily integrated into the optical package. While combinations of many these features can be found in other types of devices, the AOTF is unique in its simultaneous multiple wavelength capability. As such it can remove latency and simplify the development of recognition algo-

[1] This research has been sponsored in part by Office of Naval research (ONR) under Contract N00014–95–1–0591. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ONR or the U.S. Government.

rithms. The key components needed to incorporate optical preprocessing with a single IR camera are shown in Figure 2. Figure 3 shows a photograph of the hardware under assembly for use on this MURI project. The input or first camera lens images the scene onto the first imaging plane which contains a field-of-view defining iris. This iris is adjusted to provide complete spatial separation of the filtered and the spectrally notched images and, if necessary, can be adjusted to suppress unwanted "flare." The middle lens (also a standard camera-type lens) "collimates" the light that passes through the AOTF. A phase retarder may be added if polarimetric analysis is warranted. The third camera lens brings the filtered light from the first imaging plane onto the imaging plane of the camera. Our experience confirms that this optical arrangement is superior to alternative geometries that have been reported.

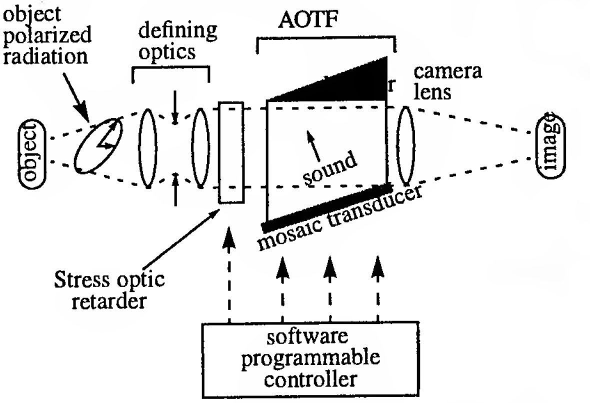


Figure 2: Key components of the multispectral imager.

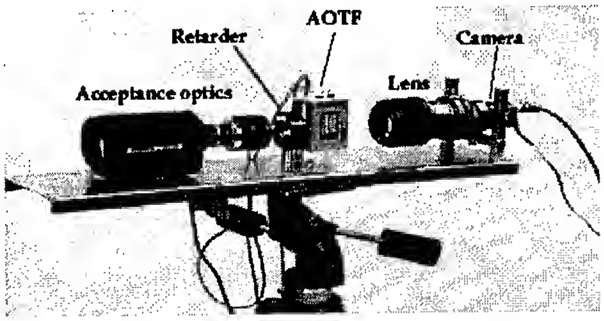


Figure 3: Smart filter camera system that will be used for data collection.

Initial experiments on multispectral imaging of a color scene containing "friend" and "foe" vehicles have provided very encouraging results as shown in Figure 4. The "signature" and "non-signature" images were taken through AOTF tuned to red (600-630 nm) and green (500-530 nm) regions of visible spectrum, respectively. We used a Visual Basic® computer code to implement pixel–by–pixel processing of the raw images. As a result of the processing, considerable reduction of the white background was achieved with simultaneous strong increase in contrast of "friend" vehicle icon image with respect to a "foe" vehicle icon and a background (multiplication). Finally, intensity thresholding of the scene yields an image of the vehicle icon of interest only and suppresses all the other features (threshold subtraction).

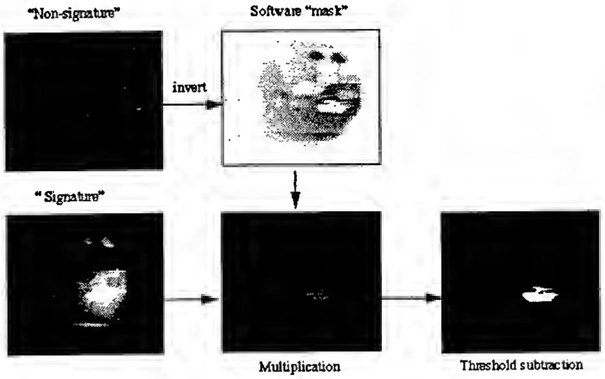


Figure 4: Multispectral filtering can simplify the processing in object recognition scenarios and enhances target identification.

3. Factors Affecting AOTF Image Quality

We have addressed one of the critical issues that has been the characterization and quantification of the various factors that impact the overall image quality. The AOTF, because of its diffractive nature, degrades image quality unless adequate measures and compensation is made. The advantages of the AOTF in spectral imaging systems are largely the same as for non-imaging systems, and have been well established by recent work. The published literature contains many examples of AOTF's being successfully incorporated in imaging systems to be used for a variety of applications, but also point to several critical issues that must be addressed. These issues are largely driven by the particular requirements for high quality imaging, and often are not relevant for non-imaging applications. It is generally necessary to pay close attention to optimizing the design of the AOTF to assure good imaging quality, and to recognize the limitations that may be imposed by the fundamental physical effects.

A variety of physical factors must be considered in high image quality AOTF spectral imaging systems. The factors are manifested both in the basic design and in the implementation of AOTF-based imaging systems. The spatial resolution, background level and other quality criteria of the AOTF spectral image depend critically on the AOTF design details. We have analyzed the factors as they relate to spectral resolution, spatial resolution and AO interaction dispersion. Fabrication issues affect suppression of background through acoustic scattering, acoustic field homogeneity and optical scattering. In this paper we will review these effects and describe some measurements and analysis that we carried out.

3.1. Image Blur

One of the major advantages of the AOTF is the relatively large angular aperture which enhances the throughput of the device. The large acceptance aperture results from the birefringent character of the diffraction process, so that there will be a dependence of diffracted image angle on wavelength; there will therefore be a blur, or angular spread, in the image due to the finite spectral bandpass of the AOTF. This angular spread can seriously degrade the image quality unless steps are taken in the design to minimize it. It may even be possible to produce practical systems in which the image resolution is no worse than imposed by aperture diffraction.

Closely related prior work on AOTF imaging blur in which one of our group had been involved was recently reported [Suhre et al., 1992]. These investigations were done in the 8 to 12 micrometer infrared range utilizing a TAS device. Those results were greatly extended under the present MURI program. Calculations were made for $\Delta\Theta_d$, the internal angular spread of the diffracted light as a function of $\Delta\lambda$, the bandpass of the AOTF. The principal design parameters for a noncollinear AOTF are shown in Figure 5. The increase in $\Delta\Theta_d$ with bandpass is expected, since for a fixed value of the angle of incidence, Θ_i, the acoustic beam spread must increase to accommodate increasing values of bandpass. This must be done by decreasing the acoustic interaction length, the only remaining design parameter. The increasing values of $\Delta\Theta_d$ with increasing Θ_i can be understood through some simplifying approximations in the analysis. For small values of $(n_i - n_d)$ and $(\Theta_i - \Theta_d)$, the non-critical phase matching (NPM) condition can be approximated as

$$\lambda_0/\Lambda = n_0(\Theta_i - \Theta_d) \tag{1}$$

where n_0 is the ordinary index of refraction.

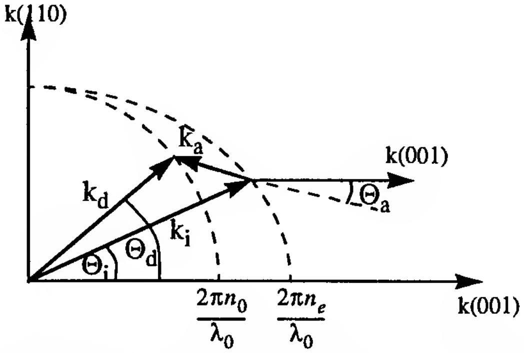


Figure 5: Principal design parameters of an AOTF.

The usual NPM approximation for the acoustic tuning wavelength, Λ, is

$$\frac{\lambda_0}{\Lambda} = \Delta n(\sin^4\Theta_i + \sin^2 2\Theta_i)^{1/2} \tag{2}$$

so that an approximation to the beamspread is

$$\frac{\Delta\Theta_d}{\Delta\lambda} = \frac{\Delta n}{n_0\lambda_0}(\sin^4\Theta_i + \sin^2 2\Theta_i)^{1/2} \tag{3}$$

This approximation agrees well with the exact calculations [Suhre et al., 1992]. While this simplified formulation shows the explicit dependence of the blur on the AOTF bandpass, it is straightforward to recast the dependence on the transducer length by substituting for $\Delta\lambda$ its dependence on transducer length Λ,

$$\Delta\lambda = \frac{1.8\pi\lambda^2}{(\Delta n L \sin^2\Theta_i)} \tag{4}$$

to obtain

$$\Delta\Theta_d = \frac{1.8\pi\lambda}{n_d L}\left(\frac{\sin^4\Theta_i + \sin^2 2\Theta_i{}^{1/2}}{\sin^2\Theta_i}\right) \tag{5}$$

It is clear that near diffraction limited performance for practical designs will only be achieved for fairly long interaction lengths L yielding high spectral resolution, since we must satisfy the condition that $\Delta\Theta_d < \lambda/D$. For a one centimeter aperture and a wavelength of 10 micrometers, this will be a beamspread of about 1 milliradian; (5) suggests that values of Θi must, in general, be kept small for good spatial resolution. The experimental measurements reported in reference 1 carried out in the wavelength range from 7 to 11 micrometers were in good agreement with this analysis.

Effects of image blur are more prominent in the visible and near IR using TeO_2 crystals, and great care is needed in designing imaging AOTF's to minimize blur effects. The two critical effects are due to bandpass, as described above, and image shift due to dispersion. Analysis and measurements were carried out at CMU [Wachman, 1996] on a TeO_2 AOTF custom built for this purpose by NEOS. Transducer length could be varied by external interconnections. The basic AOTF design parameters for this AOTF are listed in Table 1.

TABLE 1. AOTF–1 Parameter

$$\Theta_i = 12°$$

$$\Delta\theta_i = 6.5°(ext)$$

$$\Theta_a = 5.9°$$

$$L_1 = 0.33cm$$

$$L_2 = 0.66cm$$

$$L_3 = 1.32cm$$

$$L_4 = 2.32cm$$

$$\Delta\lambda/\lambda = 0.1 \text{ for } L = 2.32\text{cm}$$

The transducer structure consisted of four elements, each wired to a connector on the mount so that it could be driven independently. Analysis and measurements on image blur were carried out, and the results are summarized in Figure 6 for the four values of transducer length. The analysis indicates that a transducer length of about 3.5cm is required to reduce the image blur to no greater than that due to aperture diffraction from a 1 cm aperture, at 0.7 micrometers wavelength. This calculation is in good agreement with the experimental results in Figure 6, which shows spectrally filtered photographs of a resolution chart taken with the four transducer lengths. The horizontal bars, unaffected by AO diffraction, are near diffraction limited resolution. For L = 2.32 cm the analysis predicts the AO blur to be about 1.5x the diffraction limit, while it is about 6x for L = 0.33 cm. This AOTF design is a reasonably good match between the spectral and spatial resolution characteristics.

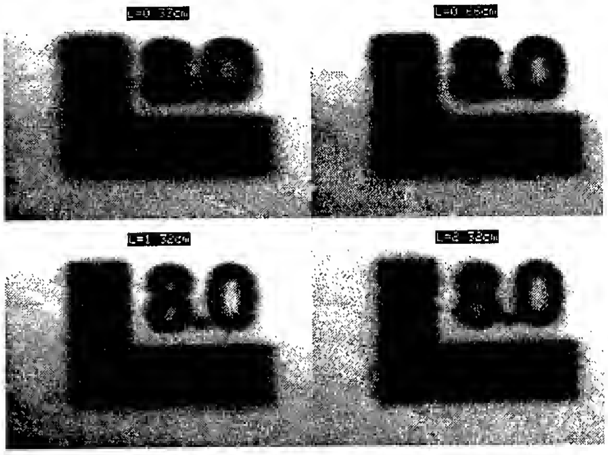


Figure 6: Target images taken with various

3.2. Background Illumination

Another major source of AOTF image degradation is the loss of contrast due to high background levels. This background is largely out-of-band wavelengths, and is principally due to three causes: high sideband levels, light diffracted by acoustic energy reflected at various crystal surfaces, and light scattered by the crystal from its bulk, surfaces, and coatings. Measurements of this spectrally broadband background were made on the AOTF described above, and the results are summarized in Figure 7.

The AOTF was tuned for peak transmission at 5 wavelengths across its range: 470 nm, 486 nm, 535 nm, 650 nm and 710 nm. A spectrometer was used to measure the intensity from 400 to 770 nm for each of these tuning conditions. Close to the main peak, high sidelobes are the principal cause of the background, while further from the main peak phase matching to reflected acoustic energy may be dominant. The latter contribution was measured by using pulsed RF power and gating the detector so as to discriminate against the diffracted light due to the first transit acoustic wave. The background due

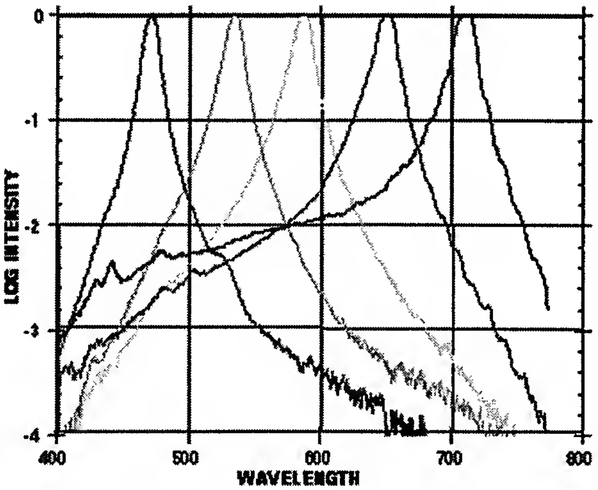


Figure 7: Spectral intensity distribution for several AOTF tuned wavelengths.

to this cause is greater than –20 dB from about 450 to 650 nm. For imaging application for which a dynamic range of more than 30 dB is needed, it is clear that steps must be taken to greatly reduce the background from these effects through proper AOTF design. We addressed the background issues with a TeO2 AOTF imaging system based on a design described in Table 2.

TABLE 2. AOTF–2 Parameters

$$\Theta_i = 10.7°$$

$$\Delta\theta_i = 6.5°(ext)$$

$$\Theta_a = 4.0°$$

$$\theta_{\text{face wedge}} = 3°$$

$$\theta_{\text{blur}} = 0.5\text{rad (ext), 3x diffraction limit}$$

$$L = 1.43\text{cm (3 elements)}$$

$$\Delta\lambda/\lambda = 0.1$$

We have found that a major cause of image degradation relates to the transducer structure and method of interconnecting elements. In order to achieve good impedance matching for a large area transducer, it is necessary to either interconnect several small elements in series, or drive each independently from a power splitter. In the latter case, the elements may have either a common ground plane or isolated grounds. For series connection a single matching circuit is used, and for parallel connection each element has its own matching circuit. The imaging results we obtained using multiple elements generally shows that there are distinct, multiple images, and a blur for each image correspond not to the entire transducer length, but only to the element length. This behavior suggests that the acoustic wave components generated at each element are not coherent, possibly not co-directional. We have found with one parallel connected AOTF, for which each element has its own matching circuit, multiple images resulted from cross-talk between the matching circuit due to inadequate isolation. In each case, we found, as expected, a strong correlation between image quality and how closely the passband of the AOTF adhere to the sinc^2 function. One such passband characteristic is illustrated in Figure 8; note the significant departure from a sinc^2 function.

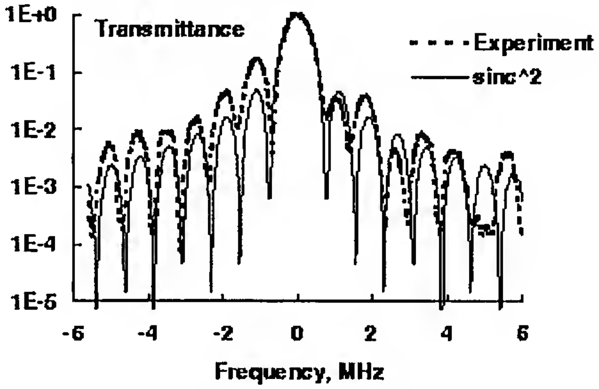


Figure 8: Spectral resolution of NEOS 4-3-P-2 AOTF.

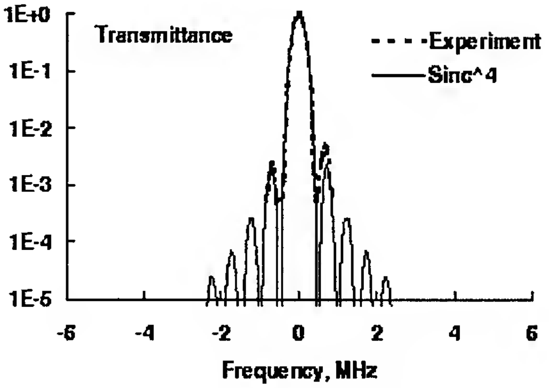


Figure 9: Spectral resolution of NEOS 4-3-P-1 and 4-3-S-1 in series

It is important to realize that even in the case of perfect passband behavior, the image will be corrupted by the sidelobes, the first of which is only 13 dB below the main signal. This level is inadequate for many imaging applications. Attempts to address this problem by apodization of the acoustic field have not been particularly successful because of the transducer fabrication difficulties in achieving a good enough approximation to the required field profile. We have demonstrated an alternative approach to sidelobe reduction which is based on the use of two AOTF's in series, for which the passband characteristic is sinc^4. The expected transmission for two identical filters in series is a reduction of the first sidelobes to –26 dB, and a reduction in the FWHM of the main lobe by to about 75% of the single filter. A measurement of the passband was made for two AOTF's in series, and the results are shown in Figure 9. The first sidelobe is about 25 dB below the main peak, in good agreement with theory.

Measurements were made of the white light spectral imaging characteristics of the same two AOTF's, individually and with both in series in the optical train. The target consisted of a cross cut into sheet metal, and illuminated from behind. The width of the cutout slit was 3 mm, and the target was placed a distance of 2.5 meters from the camera; therefore the slit subtended an angle of 1.2 milliradians at the input optics. Figure 10 shows the image of the target, and the intensity traces of the slit perpendicular to the AO diffraction plane, and parallel to the diffraction plane. The image of the vertical slit is corrupted by the blur and sidelobe effects, and this is reflected in the trace. In Figure 11 we show the image taken with the two AOTF's in series, in which the sidelobe images are no longer visible. We also show the intensity trace for the vertical slit, superposed on the trace for the single AOTF to show the same effect.

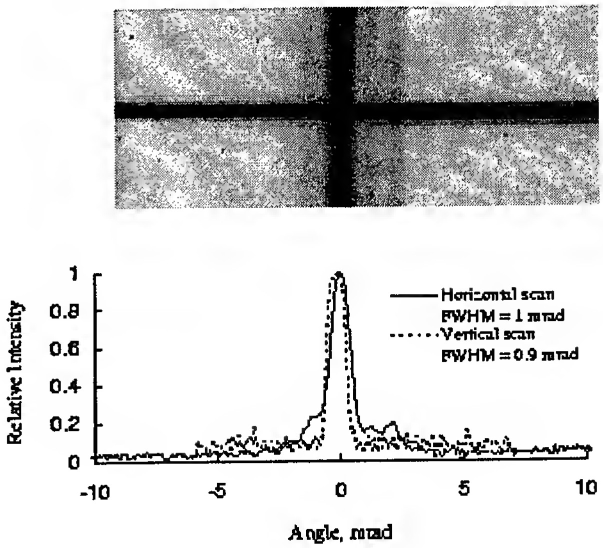


Figure 10: Image recorded with NEOS 4-3-S-1 AOTF, and intensity scans parallel and perpendicular to diffraction direction.

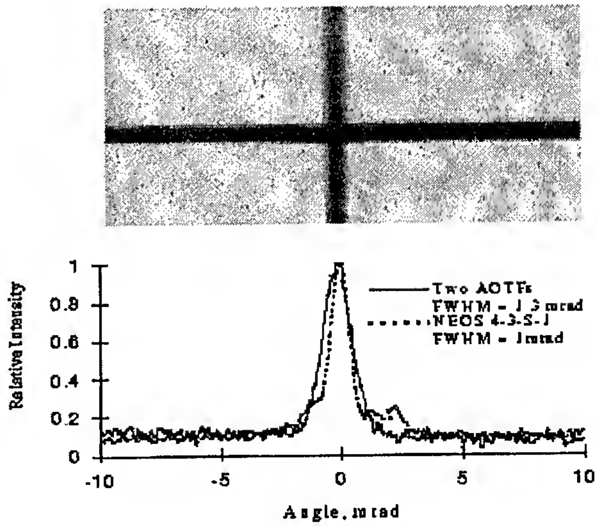


Figure 11: Image recorded with two AOTFs in series, and intensity scan compared with single

3.3. Scattering

A contribution to broad spectral background illumination limiting dynamic range will be caused by scattering of the incident white light image from the AOTF crystal volume, and its surfaces. This is not normally a problem, except for very high dynamic range systems, or very high resolution filters. A rough estimate of this limit can be made from existing scattering data, which suggests that for the best optical quality oxide crystals such as TeO_2, forward scattering values are about -50 dB; the scattering will be greater for most infrared AO crystals, such as TAS or $Hg_2\ Cl_2$. Assuming a cos^2 intensity dependence with respect to the forward direction, scattering into the diffracted image acceptance aperture is nearly the same value for typical AOTF designs. The ratio of scattered light intensity to diffracted image signal is approximately

$$\frac{I_{\text{scat}}}{I_{\text{image}}} = \frac{S}{p\eta}\cos^2\phi\frac{\Delta\lambda}{\delta\lambda} \qquad (6)$$

where S – scattering coefficent, ϕ – scattering angle, $\delta\lambda$ – AOTF resolution, $\Delta\lambda$ – spectral range of light source and detector, p – polarization loss, at least 50%, and η – AOTF efficiency.

For the AOTF design of table 2, S~10^{-5}, $(\Delta\lambda/\delta\lambda) = 100$, $p = 0.5$, and $\eta = 0.5$, so that the estimated scattered light intensity is about 24 dB below the image signal. Measurements were made of the scattering level from the experimental AOTF's using a He–Ne laser as source, and scanning the detector in the image plane. These results are in reasonably good agreement with this estimate.

4. IR Crystal Growth Activities

For AOTF applications in the visible to far (to 20 µm) IR range, mercurous chloride has very attractive properties [Barta, 1970]. However, it is not available commercially. Nevertheless, there is considerable prior knowledge in the literature originating from C. Barta of Czechoslovakia [Barta et al., 1975], and more recently from a Westinghouse Research team, led by N.B. Singh [Sing et al., 1986], about the seeded vapor growth of mercurous chloride. Our goal is to establish the equipment and the hands-on skills needed to grow mercurous chloride. During 1996, we grew two crystals from commercial mercurous chloride powder samples, one from Aldrich and the second from Fisher. Significant improvement of crystal formation was made from the first to the second growth, primarily through the acquisition of a new turbo pump vacuum system. We used a two zone furnace for this growth. Its actual temperature profile for mercurous chloride growth is shown in Figure 12. Fine adjustment is required to initiate seeding either at a random site (unseeded growth) or on a suitably oriented pre-existing crystal (seeded growth). Only the crystal grown from Fisher material has the appearence of a single christal. Both crystals are awaiting X–ray characterization. .

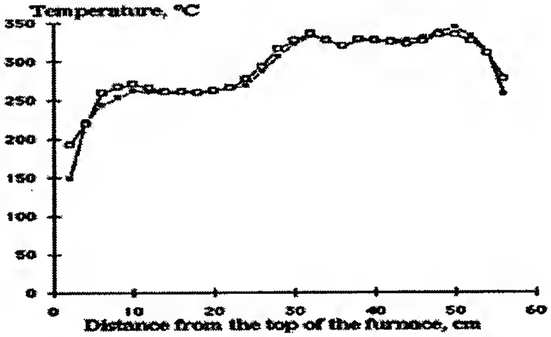


Figure 12: The temperature profile of the crystal growth furnace.

5. Future Plans

The AOTF that we designed for optimized broad spectral band operation has been built to our specifications and delivered. This device is being characterized as to its electrical and acousto-optical parameters to fully exploit its capabilities in smart filter scenarios. We will address the important issue of minimizing spectral image blur with a new device configuration in which two AOTF's are used in tandem to reduce the impact of spectral side lobes. We will then proceed with the major task towards combining the operation of the AOTF with computational sensors. In the crystal growth activities, we plan to establish seeded growth of mercurous chloride. Key tasks include setting up for crystal orienting, cutting, and polishing.

References

[Barta, 1970] C. Barta, "Preparation of Mercurous Chloride Monocrystals", *Kristall und Technik* **5**, 541–549(1970).

[Barta et al., 1975] C. Barta, J. Trnka and J. Cemlicka, "On the Nature of Calumel Vapor" *Kristall und Technik* **10**, 775–781(1975).

[Sing et al., 1986] N.B. Sing, R.H. Hopkins, R. Mazelsky, and J.J. Conroy, "Purification and Growth of Mercurous Chloride Single Crystals", *J. Crystal Growth* **75**, 173-180 (1986).

[Suhre et al., 1992] D.R. Suhre, M. Gottlieb, L.H. Taylor, and N.T. Melamed, "Spatial resolution of imaging noncollinear acousto-optic tunable filters", *Optical Engineering* **31**, 2118-21 (1992).

[Wachman, 1996] E.S. Wachman, Carnegie Mellon University Center for Light Microscope Imaging and biotechnology, private communication, 1996.

Automated Vision and Sensing Systems at Boston University *

Stephen Grossberg, Gail Carpenter, Eric Schwartz, Ennio Mingolla,
Daniel Bullock, and Paolo Gaudiano,
Department of Cognitive and Neural Systems, Boston University;
Andreas Andreou and Gert Cauwenberghs, Department of Computer Science
and Electrical Engineering, Johns Hopkins University; and
Allyn Hubbard, Department of Electrical and Computer Engineering, Boston University
email: steve@cns.bu.edu URL: http://cns-web.bu.edu/muri

Abstract

Our Center is carrying out a coordinated program to develop general-purpose autonomous systems for vision, object recognition, and control applications. The systems are realized in software, off-the-shelf hardware, and customized chips. These systems are designed to operate within noisy environments for which rules are not known and which can change unexpectedly through time. They typically begin with models of a key brain competence and end with fielded applications that have been thoroughly benchmarked. Projects include psychophysical studies of how humans search complex scenes; models of coherent processing of noisy and incomplete image data from natural and artificial sensors; development of self-organizing classifiers capable of fast, stable, distributed, incremental learning and hypothesis testing in response to nonstationary, incomplete, and probabilistic data; algorithm and hardware development for head-mounted space-variant active vision systems; development of self-calibrating autonomous robots; and fabrication of chips for vision and classification applications.

1 Introduction

This report summarizes research being conducted under the Multidisciplinary University Research Initiative (MURI) program by the Boston University Department of Cognitive and Neural Systems, the Boston University College of Engineering, and the Johns Hopkins University Department of Electrical and Computer Engineering. Our MURI Center is developing general-purpose autonomous neural systems for vision, object recognition and control applications. These systems are designed to operate within the types of uncontrolled environments that are typified by the battlefield. Such environments may contain rare but important events whose consequences differ from those of similar frequent events, as well as unexpected combinations of events, irregular statistical drifts in event sequences, and different amounts of morphological variability in objects to be detected, recognized, and controlled.

Our projects typically begin by modelling a key biological competence. These models aspire to be general-purpose solutions of modal problems, such as vision, adaptive pattern recognition, and adaptive sensory-motor control. We are guided in our model development by the typically huge psychophysical and neurobiological data bases in these fields. These data bases are the first explanatory and predictive targets of the model. For example, we developed a new model of how the visual cortex is organized into layers, columns, maps, networks, and successive processing levels to generate context-sensitive boundary segmentations of image data. Where sufficient data are not available and the competence is important, we collect and analyse the

* Supported in part by the Defense Advanced Research Projects Agency and the Office of Naval Research (ONR N00014-95-1-0409).

data ourselves. Our mathematical and computational analyses then characterize each model's functional properties. Once they are mathematically understood, the models can be modified and optimized for a wide variety of applications. Earlier versions of these models have rapidly been applied by technologists because they exhibit human-compatible properties of autonomous adaptation or performance in response to various types of changing environmental conditions.

In order to optimize the models for applications, they are tested by being benchmarked against competing approaches. Often there are no competing approaches, because various of the models have combinations of desirable self-organizing properties that have not yet been achieved elsewhere. For example, the Distributed ARTMAP algorithm summarized herein joins properties of fast, stable, distributed, and incremental learning and hypothesis testing for classification of arbitrarily large amounts of nonstationary data. This combination of properties has not elsewhere been achieved, to the best of our knowledge. After the models are successfully benchmarked, they are then realized in real-time software, off-the-shelf hardware, or custom VLSI chips.

Illustrative applications that are reviewed here include boundary and surface processing of natural and synthetic images, development of self-organizing classifiers of textured synthetic aperture radar (SAR) images, geospatial mapping from satellite remote sensing data, medical prediction in the field, radar range profile target recognition, automatic generation of coherent and attentive representations of object motion, continuous motion tracking in response to spatially and temporally discontinuous signals, fusion of form and motion data to predict object motion in noisy environments that could not be achieved using motion data only, psychophysical experiments to determine how human observers search complex scenes, adaptive multimodal fusion of visual, auditory, and planned movement commands for attentive control of ballistic movements, algorithmic and hardware development of head mounted space-variant active vision systems, navigation by self-calibrating robots under visual guidance, and the development of neuromorphic VLSI for vision and adaptive classification applications.

2 Boundary and Surface Processing of Natural and Synthetic Images

2.1 Boundary Segmentation and Surface Representation

Automatic boundary segmentation and surface reconstruction of noisy and cluttered scenes remain key problems for applications. Such preprocessing is needed for use by expert photointerpreters, by non-expert users of SAR, multispectral infrared (IR), and laser detection and ranging (LADAR) imagery in battlefield conditions, and as a preprocessor for such image data before it is automatically classified by adaptive pattern recognition algorithms. Nonparametric methods that can cope with arbitrary images are needed to deal with many battlefield situations. Our approach to boundary segmentation provides nonparametric multiple-scale data fusion and parallel decision-making algorithms whose design principles and circuit mechanisms can be generalized to many problems.

This project continues the development of boundary segmentation circuits with the following properties: (1) automatic compensation for variable illumination gradients ("discount the illuminant"), (2) suppression of noise in a context-sensitive fashion, (3) completion of boundary groupings in response to mixtures of noisy edges, textures, and shading, (4) completion of noise-free surface representations using the outputs of (1)–(3). The resulting algorithm has been tested, for example, on SAR imagery of wooded scenes with man-made roads from MIT Lincoln Laboratory. The SAR images were obtained using a 35-GHz synthetic aperture radar with 1 foot by 1 foot resolution and a slant range of 7 km [Novak *et al.*, 1990]. Earlier versions of the algorithm have been used by Lincoln Lab, among others, to preprocess SAR, multispectral IR, and LADAR images, and similar technology transfers are anticipated for the new algorithm. The new circuits are simpler computationally, run faster, and provide better noise suppression, boundary localization, and boundary completion properties. These circuits form the backbone of a larger circuit under development which also (5) separates objects from each other and from their backgrounds in 3-D, and (6) completes representations of partially occluded objects in response to both 3-D scenes and 2-D pictures (see Section 4). These boundary and surface circuits form the front end of architectures that include self-organizing pattern recognition algorithms for incrementally learning to classify scenic objects, textures, and image understanding interpretations (see Sections 3 and 5).

2.2 Boundary Segmentation: From Visual Cortex to IU Algorithm

Sources for our new IU designs are the huge experimental literatures on human visual psychophysics and primate neuroscience. Understanding how the visual cortex is organized to yield the properties of visual perception is one of the outstanding questions in the science and technology of vision. The visual cortex is organized into layers, columns, maps, networks, and successive processing stages. One project is developing a computational model of how all these structures are organized for purposes of boundary segmentation in the first three processing stages (lateral geniculate nucleus (LGN) and interblob cortical areas V1 and V2) beyond the photosensitive retina. This model suggests how the visual cortex elegantly accomplishes the image processing goals (1)–(3) stated above [Grossberg, Mingolla, and Ross, 1997] using compact and modular circuits. These circuits provide functional explanations for identified cortical cells and connections, and have simulated challenging psychophysical and neurophysiological data. The circuits have also been used to successfully process SAR data, and will be incorporated into the next generation of vision chips from our Center's VLSI (very large scale integrated circuit) team.

A schematic circuit diagram of this FACADE network is given in Figure 1. Key design features are summarized because of the breakthrough nature of these results. Neural labels are used for definiteness: (1) The circuit retains analog sensitivity to distributed image features, even as it generates coherent and context-sensitive boundary segmentations. (2) Several types of feedback circuits, which equilibrate very quickly (in 1–3 cycles), ensure this analog sensitivity. (3) One type of feedback circuit operates within each brain region. It generates the groupings whereby edge, texture, shading, and stereo information are bound together into coherent segmentations. Such circuits use direct long-range horizontal cooperation and two-stage (i.e., disynaptic) short-range competition within the complex cells of cortical layer 3 to realize a bipole property (see Figure 2A). The bipole property enables a complex cell to fire when it lies between nearly aligned inducing signals, but not when it lies beyond a single inducer. (4) Another type of cooperative and competitive interaction works with the bipole property to generate context-sensitive boundary groupings. Excitatory inputs from LGN arrive in area V1 at layers 4 and 6 (Figure 2B). LGN inputs directly excite orientationally tuned simple cells in layer 4. In particular, oriented arrays of spatially displaced LGN ON and OFF cells excite mutually inhibitory simple cells that are

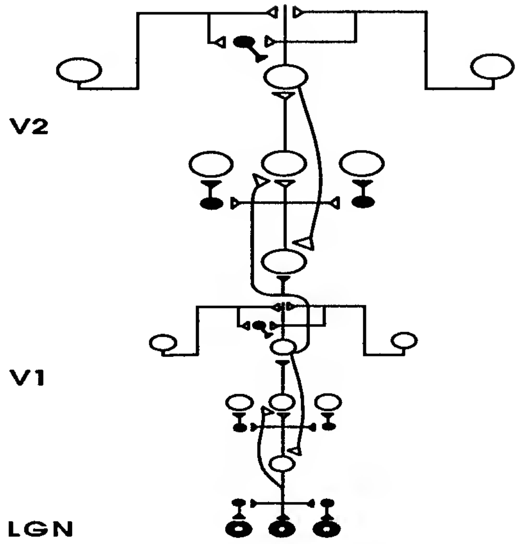


Figure 1: Schematic of LGN–V1–V2 model circuitry. The V2 circuit replicates the V1 circuit but at a larger spatial scale. Open symbols = excitatory, closed symbols = inhibitory.

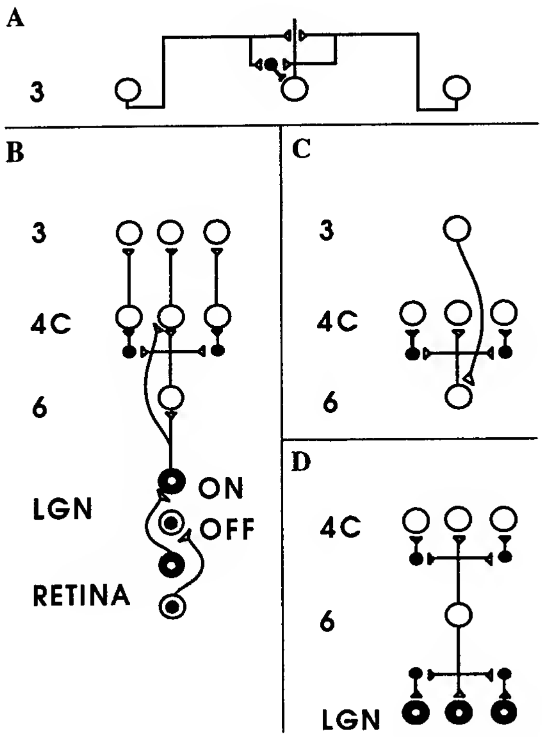


Figure 2: Model retinal, V1, and LGN circuit. See text for details.

sensitive to the same orientation but opposite contrast polarities (not shown in Figure 2). After layer 6 cells are activated, they, in turn, both excite and inhibit layer 4. The net effect is that LGN influences layer 4 via a feedforward on-center off-surround network of cells (Figure 2B) that obey membrane, or shunting, equations. This excitatory-inhibitory balance enables layer 4 simple cells to maintain their analog sensitivity to visual inputs of variable contrast.

The next interactions close the feedback loop that accomplishes boundary segmentation: (6) Layer 4 cells activate cells in layer 3 (Figure 2B), which then attempt to cooperate using their long-range horizontal connections and short-range disynaptic inhibition. All activated cells feed back to layer 6 (Figure 2C). Layer 3 hereby gains access to the on-center off-surround network of connections from layer 6 to layer 4. (7) The long-range bipole grouping in layer 3 can use the shorter-range layer 6-to-4 signals to amplify those cell activations that are favored by bipole grouping and suppress those that are not, while maintaining analog sensitivity. Layer 6-to-4 inhibition influences different orientations and positions by being distributed across a data structure (called a hypercolumn map) wherein cells sensitive to these features are spatially organized. Using this data structure, the short-range competition can relatively enhance cell responses cooperating in positional, orientational, and length-sensitive groupings by suppressing cells responding to weaker groupings, incoherent noise, or background signals. Thus, the same circuit that maintains sensitivity to bottom-up inputs also does so in response to top-down signals, and in so doing closes the fast feedback loop that generates boundary segmentations.

(8) The second type of feedback circuit operates between successive brain regions. For example, after LGN activates layers 4 and 6 in area V1, feedback from layer 6 in area V1 to LGN selects and synchronizes LGN activities that are consistent with cortical cell activity, and suppresses all other LGN cells. This feedback quickly focuses attention upon the selected LGN cells and increases the visual information transmitted from LGN to cortex. Thus, the same feedback signals from layer 3 to 6 that start to generate segmentations in V1 also selectively enhance the LGN inputs that will support the segmentation process. We have proposed that similar attentional circuits operate at all levels of the visual cortex, including areas V2 and V4, in order to explain neurophysiological data from these areas. We also predict that these top-down attentional circuits stabilize the learning process whereby bottom-up adaptive filters autonomously tune their parameters in response to changing input statistics.

(9) Another efficient design links areas V1 and V2. The model proposes that V1 and V1 are self-similar copies of one another, but that V2 has longer-range interactions than V1. Various data support this hypothesis. Groupings across the smaller scales in V1 enhance responses that code for mutually consistent boundary locations and orientations, while larger-scale groupings in V2 supports long-range boundary completion across textures and large pixel drop-outs, which in the brain are due to the blind spot and veins

of the retina. Layer 3 in V1 activates layers 4 and 6 in V2, much as LGN activates V1.

2.3 Enhancement of Synthetic Aperture Radar Images

The boundary segmentation system is part of a Boundary Contour System (BCS). The BCS interacts with a surface representation system that is called the Feature Contour System (FCS). Both the BCS and the FCS receive input signals only after the illuminant has been discounted by a shunting on-center off-surround network. BCS boundaries block the diffusion of discounted inputs to FCS compartments. This version of anisotropic diffusion [Cohen and Grossberg, 1984; Grossberg and Todorović, 1988] can achieve more precise surface representations than schemes which do not fully exploit the fact that boundaries and surfaces are formed using complementary processing rules [Grossberg, Mingolla, and Todorović, 1989]. In particular, FCS diffusion is restricted not only by the presence of high intensity gradient values in the original image, but also by long-range colinear (or nearly colinear) groupings of image gradient signals. These coherent boundaries determine which fluctuations in pixel intensity will be defined as noise and smoothed over, and which will be enhanced as image structures. This coherence property becomes especially important in the presence of the high noise, pixel drop out, and incomplete image data that may characterize battlefield conditions.

Our first application of the new BCS/FCS model was used to enhance images of range data from a SAR sensor. The model was used to make structures such as motor vehicles, roads, and buildings more salient to human observers than they are in the original imagery, thereby making SAR imagery, and related types of images useful in battlefield situations, to individuals without extensive training as photointerpreters. The shunting network performs a local normalization of dynamic range in one processing step. The new BCS algorithm generates positionally more accurate boundaries with significant reductions in processing time and algorithmic complexity over previous BCS circuits. The BCS-gated diffusion process in the FCS reduces speckle noise, smoothes image brightness in a form-sensitive way, and enhances contrast-differences between different image surfaces (see Figure 3). The BCS/FCS algorithm outperforms alternative published algorithms for image enhancement, as detailed for an earlier version of the algorithm in Grossberg, Mingolla, and Williamson [1995]. For example, the BCS/FCS algorithm typically converges in 1–3 iterations to a stable image reconstruction as its algorithm is iterated. In contrast, the correct number of iterations of median, sigma, and geometric filters is image-dependent. The new BCS algorithm runs in approximately 20% of the time required for the published algorithm of Grossberg, Mingolla, and Williamson [1995] on comparable hardware. Dr. Allen Waxman's group had previously reported processing times of approximately 50 seconds on DEC Alpha stations for multiscale processing of large (512×512) SAR images, using a simplified and optimized form of our earlier algorithms [Waxman *et al.*, 1993]. Processing times of about 10 seconds per image for personal workstations thus appear realistic for our present algorithm with a comparable optimization effort. Earlier versions of this technology have been applied and developed by Waxman's group for military applications. His work is now funded under the Integrated Imagers Initiative of DARPA/ETO.

3 Gaussian ARTMAP and ARTEX Classifiers

3.1 Gaussian ARTMAP versus Expectation Maximization Classifiers

The BCS boundary and FCS surface representations are often used to generate output vectors that are input to a self-organizing pattern categorizer, or classifier. Such a system can autonomously classify data from multiple sensor types, or can aide a human observer in his/her classification performance. Many projects within our Center are developing Adaptive Resonance Theory (ART) classifiers, because they combine a series of properties that are of importance in battlefield situa-

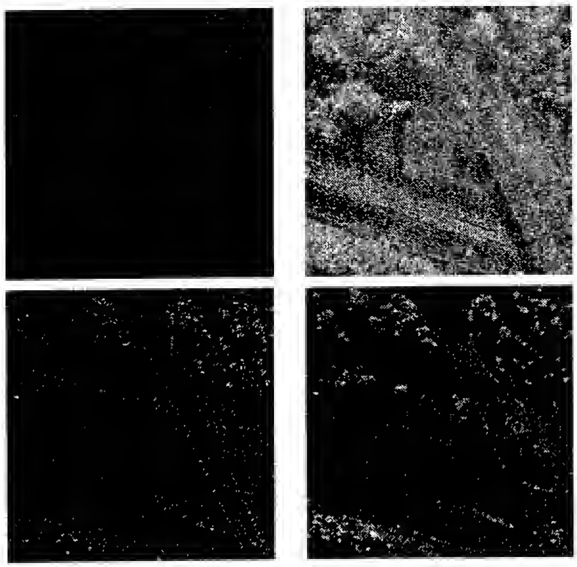


Figure 3: (A) Top left: Unprocessed SAR image of upstate New York scene consisting of highway with bridge overpass. (B) Top right: logarithm-transformed SAR image. (C) Bottom left: Center-surround, shunting network result averaged across spatial scales. (D) Bottom right: New BCS/FCS multi-scale enhancement.

tions. These include the ability to incrementally learn stable recognition categories in response to an essentially unlimited number of rare events, unexpected events, irregular statistical drifts in event sequences, and different amounts of morphological variability in the events to be classified. Present research is focusing on how to simultaneously realize the constraints of fast, stable, and distributed incremental learning in response to an arbitrary nonstationary enironment of arbitrary finite size. All other algorithms known to us fail on one of more of these criteria. See Section 5 for further details.

The present project developed a new ART classifier with which to process textured SAR scenes after they are preprocessed by the BCS/FCS model. This classifier is called Gaussian ARTMAP, or GAM. We subsequently showed that this classifier outperforms state-of-the art statistical, rule-based, and multilayer perceptron algorithms on a number of other benchmark data bases, including: letter image recognition [Frey and Slate, 1991], Landsat satellite image segmentation [Feng *et al.*, 1993], speaker independent vowel recognition [Deterding, 1989], and natural texture databases [Brodatz, 1966], in addition to obtaining accurate, high resolution classification of image regions in response to BCS/FCS processed SAR data [Grossberg and Williamson, 1997; Williamson, 1996, 1997].

As with other ART algorithms, GAM incrementally self-organizes an internal categorization of its inputs, and maps those categories to output class predictions. GAM differs from other ART networks by using internal category nodes that have Gaussian receptive fields. That is, each GAM category defines a Gaussian distribution in the input space, with a mean and variance in each input dimension, as well as an overall *a priori* probability. GAM's activation function evaluates the posterior probability that the input belongs to a category, while GAM's match function evaluates how well the input fits the category's distribution. Match is a measure of the distance, in units of standard deviation, between the input vector and the category's mean. The network's vigilance parameter specifies the maximum allowable size of this distance.

The original GAM network used choice learning, in which only the maximally activated category would learn [Williamson, 1996]. GAM now uses distributed learning, in which each category is assigned credit based on its proportion of the net category activation [Williamson, 1997]. The net activation is determined by all categories that exceed vigilance and that belong to the correct *ensemble*. An ensemble consists of the set of categories that map to the same output prediction. When an input is presented, the maximally activated ensemble is chosen. If its prediction is correct, the ensemble's categories learn. If its prediction is incorrect, then *match tracking* takes place: the categories in the chosen ensemble are reset, or deactivated, and vigilance is raised. This triggers a search which continues until a correct or new ensemble is chosen.

Distributed GAM uses fewer categories and obtains higher classification rates than choice-learning GAM. We have also shown that distributed GAM is a constructive, incremental variant of an Expectation Maximization, or

EM, algorithm for mixture modeling and classification [Dempster, Laird, and Rubin, 1977; Ghahramani and Jordan, 1994]. This batch EM algorithm learns a Gaussian/multinomial mixture model, with a locally maximized likelihood, of the joint input/output space. EM then generates output predictions based on this model. We have compared GAM and EM on three real-world classification databases: letter image recognition, Landsat satellite image segmentation, and speaker independent vowel recognition. GAM outperforms EM, as well as a wide range of neural network and machine learning classifiers, on all three problems. Table 1 summarizes the best results obtained by GAM and EM.

In addition to its higher accuracy, GAM also possesses the practical advantages of constructive, incremental learning. For example, obtaining the best results on all three problems listed in Table 1 requires a wide range in the number of internal categories. With EM, a user needs to choose this number through trial and error for each problem. GAM, on the other hand, automatically constructs the appropriate number of categories for each problem. Because GAM learns incrementally, it can also be applied to situations that require immediate updating of input/output mappings as data are received, including the learning of arbitrary new contingencies, such as the addition of new class labels.

3.2 ARTEX Incremental Classification of Natural and SAR Textured Scenes

Another research accomplishment is the development of ARTEX, a self-organizing system that learns to recognize textured scenes [Grossberg and Williamson, 1997]. The ARTEX architecture, depicted in Figure 4 uses a simplified BCS/FCS network whose 17-dimensional output vectors are fed into the GAM network, which incrementally learns class predictions. ARTEX replaces the feedback loops of the BCS by a one-shot, fast, feedforward approximation that is 75 times faster than the full BCS. This approximation loses some of the photorealism achieved by the full BCS/FCS algorithm for purposes of human use, but achieves faster machine classification on natural and SAR textures without a loss of accuracy. In particular, ARTEX uses part of the BCS to extract context-sensitive texture features. This Static Oriented Contrast, or SOC, filter is computed at four orientations and spatial scales (Stages 1–5 in Figure 4). One of these scales is used to generate a surface representation (at Stage 9 in Figure 4) by gating diffusion of signals from the center-surround network (at Stage 1) that discounts the illuminant. The 16-dimensional texture feature and 1-dimensional surface brightness feature form the input vector to GAM. The OV and OI options in Figure 4 are orientationally variant and invariant pathways to the classifier, respectively, which are used in different classes of applications. We have exhaustively benchmarked variations of these input vectors.

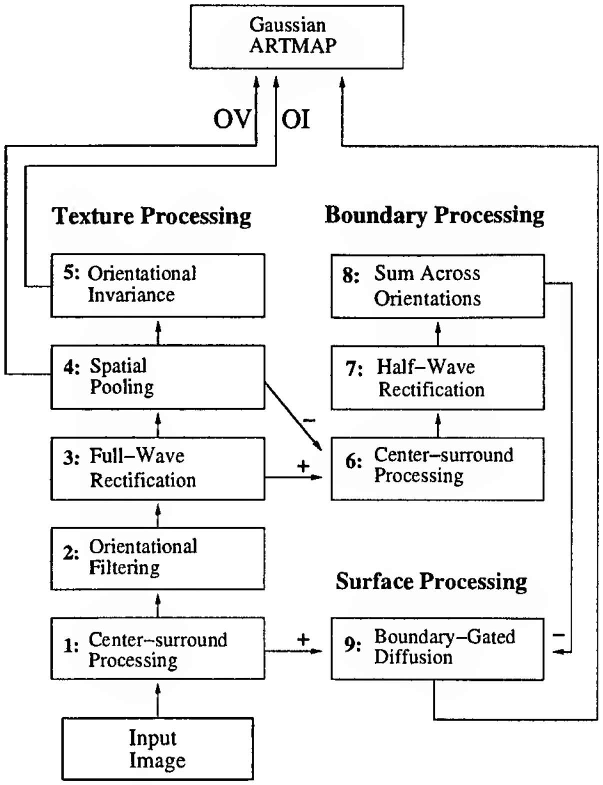

Gaussian ARTMAP
OV
OI
Texture Processing
Boundary Processing
5: Orientational Invariance
4: Spatial Pooling
3: Full–Wave Rectification
2: Orientational Filtering
1: Center–surround Processing
Input Image
8: Sum Across Orientations
7: Half–Wave Rectification
6: Center–surround Processing
Surface Processing
9: Boundary–Gated Diffusion
+
−
+
−

Figure 4: ARTEX model. See text for details.

We have also compared ARTEX to a recent state-of-the-art algorithm called the Hybrid System, which is a hybrid architecture that uses a log-Gabor pyramid for feature extraction followed by one of three alternative classifiers

Comparison of GAM and EM

Classification	GAM		EM	
Benchmark	Accuracy	# Categories	Accuracy	# Categories
Letter Images	94.6%	941	92.6%	1,000
Landsat Images	90.0%	255	89.4%	275
Spoken Vowels	56.0%	54	54.6%	40

Table 1: Results obtained by GAM and EM on three real-world classification problems.

[Greenspan *et al.*, 1994; Greenspan, 1996]. Table 2 shows the results of ARTEX and the Hybrid System on comparable benchmarks of high resolution classification of 30 natural textures. ARTEX outperforms all variations of the Hybrid System, which include a rule based classifier (ITRULE), a multilayer perceptron (MLP), and a K-nearest neighbor classifier (K-NN). We have expanded the benchmarks to include classification of up to 42 textures, and found that ARTEX scales up gracefully, with only a slight decrease in accuracy and a slight increase in memory per texture, as the number of textures increases. ARTEX maintains well over 90% accuracy even on the 42-texture benchmark.

GAM shares the advantages of the three classifiers used in the Hybrid System without their serious drawbacks. Like ITRULE, GAM predicts the posterior probabilities of the output classes. However, GAM uses a simple, incremental learning procedure, as opposed to ITRULE's complex and computationally expensive batch learning procedure. Like K-NN, GAM learns local mappings quickly. However, GAM also achieves significant data compression, unlike K-NN. MLP also achieves significant data compression, but it learns very slowly, requiring 100 times more training epochs on the 30-texture benchmark. In addition, MLP uses a form of mismatch learning that is susceptible to "catastrophic forgetting" if it is trained on new data with different contingencies from previous data.

4 FACADE Model of 3-D Vision and Figure-Ground Separation

One of the most challenging problems in vision concerns how the brain automatically separates objects from each and their backgrounds in response to static 2-D pictures and 3-D scenes. Such preprocessing is especially important for IU of partially occluded objects. Recognition is far easier in response to preprocessed images that separate occluding from occluded objects and complete the boundary and surface representations of the latter. Such completion is often said to be amodal because it influences recognition of a partially occluded object without causing a corresponding visible percept.

We have been progressively developing a computational model of how the visual cortex achieves this competence, and have used it to explain many psychophysical and neurobiological data about 3-D vision [Grossberg, 1994, 1997; Grossberg and McLoughlin, 1997; McLoughlin and Grossberg, 1997]. This is called a FACADE model because it generates a representation of Form-And-Color-And-DEpth. The FACADE model generalizes BCS boundary segmentations and FCS surface representations to the case of 3-D vision. The model interprets data from the parallel interblob (boundary) and blob (surface) processing streams that begin in the LGN and pass through cortical areas V1, V2, and V4. Output vectors from the FACADE model to a pattern classifier represent completed representations of overlapping occluding and occluded objects lying on separable boundary and surface representations that represent different depths from an observer.

Figure 5 shows a macrocircuit of this architecture. In this macrocircuit, left eye (or camera) and right eye (or camera) monocular preprocessing stages (MPL and MPR) in the LGN send parallel pathways to the BCS (boxes with vertical lines, designating oriented responses of

Comparison of ARTEX and Hybrid System

Configuration	Accuracy	# Samples/Class	#Epochs	# Categories	# Weights
Hybrid System, ITRULE	80.0%	300	Batch	—	—
Hybrid System, MLP	89.6%	300	500	60	2,700
Hybrid System, K-NN	82.0%	300	1	9,000	144,000
ARTEX	92.3%	300	5	214	7,697
ARTEX	94.4%	768	2	374	13,478

Table 2: Results of ARTEX and the Hybrid System on comparable 30-texture benchmarks.

multiple scales) and the FCS (boxes with three pairs of circles, designating opponent colors). The monocular signals BCS$_L$ and BCS$_R$ activate oriented simple cells which, in turn, activate bottom-up pathways, labeled 1, to generate a multiple-scale binocular boundary segmentation. The binocular segmentation generates output signals to the monocular filling-in domains, or FIDOs, of the FCS via pathways labeled 2. This interaction captures monocular FCS signals that are consistent with the binocular BCS boundaries and lifts them into depthful surface representations. All other FCS signals are suppressed. Reciprocal FCS-to-BCS interactions enhance the boundaries that successfully created filled-in surfaces and suppresses boundaries corresponding to more distant surfaces. This feedback loop achieves boundary-surface consistency. In so doing, it realizes an asymmetry between the processing of near and far objects that helps to explain how occluding objects pop-out. The surviving FCS signals activate the binocular FIDOs via pathways 3, where they interact with an augmented binocular BCS segmentation to fill-in a multiple-scale FACADE representation, which represents visible percepts of 3-D surfaces. This representation is compared with data about cortical area V4. The model explains how these functional properties emerge automatically from network interactions.

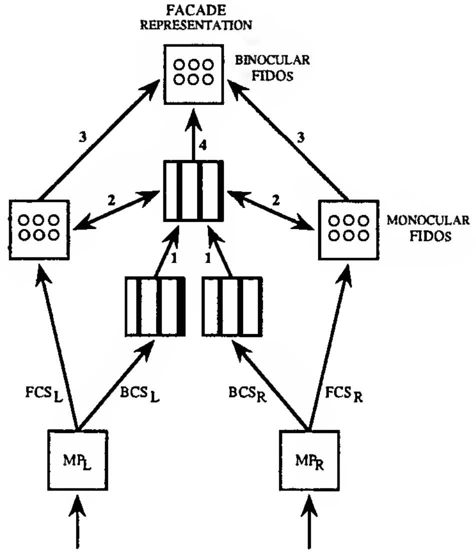


Figure 5: A FACADE macrocircuit. See text for details.

Recent projects have been computationally developing the model using psychophysical and neurobiological data. As with our other biologically-based vision projects, this will continue until the architecture is mature enough to be transferred to military and other high technology applications. As illustrated by the cortical BCS model above, technology transfer can occur very rapidly with these models as soon as they are completely characterized computationally. The latest projects have explained data about: (1) How human observers achieve binocular fusion in response to ambiguous binocular stimuli, such as Panum's limiting case, dichoptic masking stimuli, and half-occluded images in which only one eye may detect part of a scene due to its 3-D layout (daVinci stereopsis) [Grossberg and McLoughlin, 1997; McLoughlin and Grossberg, 1997]. Other computational vision models have failed to explain how humans assign the correct depth to half-occluded objects that are seen with only one eye. This project has further developed the model's multiple-scale binocular filter. The filter clarifies how humans match spatially disparate image features to the two eyes which have the same contrast polarity,

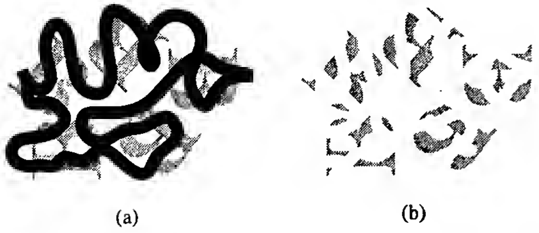


Figure 6: Relative contrasts influence pop-out and completion of partially occluded figures.

yet pool signals from both polarities in order to build effective boundary segmentations.

(2) How human observers see 2-D textures pop-out into 3-D representations that greatly influence the discriminability of different texture regions [Grossberg and Pessoa, 1997]. Of particular interest is the asymmetric effect of background luminance on whether colored texture elements are discriminable, an important issue for the design of effective displays that contain multiple sources of information. This project has led to a refinement of the model's feedback loops between the FCS and the BCS (pathways 2 in Figure 5). These pathways ensure that 3-D boundary and surface representations are computationally consistent, even though they are computed by circuits that obey complementary processing rules. (3) How changes in the the geometry of an image and in the relative contrasts of its regions interact through the cooperative and competitive processes that determine how the image will be parsed by the brain into separate objects [Grossberg, 1997]. This project has clarified how the brain uses the image T-junctions at which occluding objects intersect occluded objects to separate them from one another. The model shows how this can be done without assuming that the brain contains explicit T-junction detectors. Rather, the contextual balance of boundary cooperation and competition strengthens some boundaries while breaking others. In particular, the boundary of the stem of the T gets broken from its top as an early step in figure-ground pop-out. This explanation clarifies why models that have depended upon T-junction detectors to explain figure-ground pop-out have fallen into difficulty when confronted with many scenes.

Figure 6 is a famous demonstration of the interaction between geometry and contrast that is due to Bregman [1981] and Kanizsa [1979]. This demonstration shows that the existence of a fixed set of T-junctions in an image does not determine how it will be parsed. Instead, reversing the relative contrasts of occluding and occluded objects, without changing their T-junctions, can greatly influence how well occluded objects are completed and recognized. Figure 6B showing the limiting case in which the occluder contrast is less than that of the occluded gray fragments, indeed equals the luminance of the background, and thereby greatly reduces recognition of the same gray fragments that are easily grouped and recognized as letters B when they have a smaller contrast than the black occluder in Figure 6A.

5 ART and ARTMAP Neural Networks for Applications: Self-Organizing Learning, Recognition, and Prediction

ART and ARTMAP neural networks have been applied to a variety of problems, illustrated here by satellite remote sensing, radar identification, and medical database examples. A new family of distributed ART models retain stable coding, recognition, and prediction while allowing arbitrarily distributed category representation during learning and performance.

5.1 Technology Transfer: ART and ARTMAP Neural Networks

Researchers at the Boston University Department of Cognitive and Neural Systems/Center for Adaptive Systems (CNS/CAS), supported by basic research funding from DARPA and ONR, have introduced and analyzed the ART (Adaptive Resonance Theory) family of neural network architectures for self-organizing category learning, recognition, and prediction. Capabilities of these systems include stable incremental learning, if-then rule extraction, and database interpretation. In analyzing large non-stationary input streams, ART systems realize a combination of properties that are not shared by other neural network, artificial intelligence, or statistical methods. This research program is now advancing state-of-the-art engineering, moving from neural network models to application prototypes and fielded systems. ART networks are being used for airplane design and manufacturing, adaptive system software, circuit design speech recognition, financial prediction, medical imaging, database analysis, robotics, and defense intelligence. This technology transfer has been accelerated by consultations between development engineers and Boston University faculty researchers, as well as by the contributions of DARPA- and ONR-funded Ph.D. students and postdoctoral fellows and by CNS graduates who are implementing ART systems as part of their commercial and government employment. Examples of applications that have been published include: **a Boeing parts design retrieval system** [Caudell, Smith, Escobedo, and Anderson, 1994]; **an autonomous vision system** [Caudell and Healy, 1994]; **robot sensory-motor control** [Bachelder and Waxman, 1994; Baloch and Waxman, 1991; Bachelder, Waxman, and Seibert, 1993; Dubrawski and Crowley, 1994a]; **robot navigation** [Dubrawski and Crowley, 1994b; Racz and Dubrawski, 1995]; **active vision** [Srinivasa and Sharma, 1996]; **3-D object recognition** [Seibert and Waxman, 1992]; **face recognition** [Seibert and Waxman, 1993]; **medical imaging** [Soliz and Donohoe, 1996]; **satellite remote sensing** [Baraldi and Parmiggiani, 1995; Gopal, Sklarew, and Lambin, 1993]; **Macintosh operating system software** [Johnson, 1993]; **automatic target recognition** [Bernardon and Carrick, 1995; Koch, Moya, Hostetler, and Fogler, 1995; Rubin, 1995; Waxman *et al.*, 1995]; **electrocardiogram classification** [Ham and Han, 1996; Suzuki, 1995]; **air quality monitoring** [Wienke, Xie, and Hopke, 1994]; **weather prediction** [Soliz and Caudell, 1996]; **strength prediction for concrete mixes** [Kasperkiewicz, Racz, and Dubrawski, 1995]; **signature verification** [Murshed, Bortolozzi, and Sabourin, 1995]; **decision making and intelligent agents** [Ruda and Snorrason, 1996]; **document retrieval** [MacLeod and Surkan, 1992; Varma, Woods, and Agogino, 1996]; **analysis of musical scores** [Gjerdingen, 1990]; **character classification** [Gan and Lua, 1992; Kim, Jung, Kim, and Kim, 1995; Wang, Xu, and Ziaoliang, 1992]; **machine condition monitoring and failure forecasting** [Choi, Ly, Healy, and Smith, 1996; Ly and Choi, 1994; Tarng, Li, and Chen, 1994; Tse and Wang, 1996]; **chemical analysis from UV (ultraviolet) and IR (infrared) spectra** [Wienke, 1993, 1994]; **multi-sensor chemical analysis** [Whiteley, Davis, Mehrotra, and Ahalt, 1996]; **combinatorial optimization** [Burke, 1994]; **detection of cancerous cells** [Murshed, Bortolozzi, and Sabourin, 1996]; **sorting of recycled materials** [Wienke and Kateman, 1994]; **frequency selective surface design for electromagnetic system devices** [Christodoulou, Huang, Georgiopoulos, and Liou, 1995]; and **digital circuit design** [Kalkunte, Kumar, and Patnaik, 1992].

Ongoing basic research at Boston University continues to expand capabilities of this class of systems while collaborative projects help transfer the technology. Some of this work will now be outlined.

5.2 ART and ARTMAP Neural Networks: Introduction

Adaptive resonance theory originated from an analysis of human cognitive information processing and stable coding in a complex input environment [Grossberg, 1976, 1980]. An evolving series of ART neural network models have added new principles to the early theory and have realized these principles as quantitative systems that can be applied to problems of category learning, recognition, and prediction. Each ART network forms stable recognition categories in response to arbitrary input sequences with either fast or slow learning regimes. The first ART model, ART 1 [Carpenter and Grossberg, 1987a], was an unsupervised learning system to categorize binary input patterns. ART 2 [Carpenter and Grossberg, 1987b] and fuzzy ART [Carpenter, Grossberg, and Rosen, 1991] extend the ART 1 domain to categorize analog as well as binary input patterns [Carpenter and Grossberg, 1991]. Supervised ART architectures, called ARTMAP systems, self-organize arbitrary mappings from input vectors, representing features such as geospatial spectral values and terrain variables, to output vectors, representing predictions such as vegetation classes or environmental variables. Internal ARTMAP control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error in an on-line setting. Binary ART 1 computations are the foundation of the first ARTMAP network [Carpenter, Grossberg, and Reynolds, 1991]. When fuzzy ART replaces ART 1 in an ARTMAP system, the resulting fuzzy ARTMAP architecture [Carpenter, Grossberg, Markuzon, Reynolds, and Rosen, 1992] rapidly learns stable mappings between analog or binary input and output vectors.

5.3 Match-Based Learning versus Error-Based Learning

A *match-based* learning process is the basis of ART stability. Match-based learning allows memories to change only when attended portions of the external world match internal expectations, or when something completely new occurs. When the external world fails to match an ART network's expectations or predictions, a search process selects a new category, representing a new hypothesis about what is important in the present environment. Match-based learning, with its intrinsic stability feature, makes ART and ARTMAP well suited to problems that require on-line learning of a large and evolving database. On the other hand, error-based learning is more naturally suited to other classes of problems, such as the learning of sensory-motor maps, that require slow adaptation to statistical averages rather than the construction of a complex knowledge system. Error-based learning responds to a mismatch by sending the difference between a target output and an actual output toward zero, rather than by initiating a search for a better match. Neural networks that employ error-based learning include multi-layer perceptrons [Rosenblatt, 1958, 1962] such as back propagation [Rumelhart, Hinton, and Williams, 1986; Werbos, 1974]. ART and ARTMAP networks feature winner-take-all (WTA) competitive coding, which groups inputs into disjoint recognition categories. Other neural network learning systems such as back propagation feature distributed coding, which can provide good noise tolerance and code compression but which typically requires slow learning. Fast learning tends to cause catastrophic forgetting in these networks, as it does in ART and ARTMAP networks in which the code representation is distributed. On the other hand, fast learning is often desirable for on-line adaptation to rapidly changing circumstances and for encoding of rare cases and large databases. Variants of the basic ART and ARTMAP networks can acquire some of the advantages of distributed coding while maintaining fast learning capability. For example, ART-EMAP uses WTA codes for learning and distributed codes for testing. Distributed prediction can significantly improve ARTMAP performance, especially when the size of the training set is small [Carpenter and Ross, 1993, 1995; Rubin, 1995]. In medical database prediction problems, which often feature inconsistent training input predictions, ARTMAP-IC (instance counting) [Carpenter and Markuzon, 1996] improves performance with a combination of distributed prediction, category instance counting, and a new match tracking search algorithm. A voting strategy further improves prediction by training the system several times on different orderings of an input set. Voting, instance counting, and distributed representations combine to form confidence estimates for competing predictions. However, since these and most other ART and ARTMAP variants use WTA coding during learning, they do not solve problems such as category proliferation with noisy training sets, unless learning is slow. A new class of ART and ARTMAP networks permit fast distributed learning as well as performance. These distributed ART (dART) and distributed ARTMAP (dARTMAP) systems [Carpenter, 1997] are now being analyzed and developed for future applications. The following sections describe these developments.

5.4 ARTMAP Architecture

ARTMAP networks self-organize mappings from input vectors, representing features such as patient history and test results, to output vectors, representing predictions such as the likelihood of an adverse outcome following a procedure. Fuzzy ARTMAP incorporates two fuzzy ART modules, ART$_a$ and ART$_b$, that are linked by a *map field* F^{ab}. Many applications of supervised learning systems such as ARTMAP are classification problems, where the trained system tries to predict a correct category given a test set input vector. A prediction might be a single category or distributed as a set of scores or probabilities. A fuzzy ARTMAP algorithm, publicly available from the CNS department, outlines a procedure for these problems, which do not require the full architecture. The algorithm implements a fuzzy ARTMAP network that is a simplified version of the full network but that nevertheless is sufficient for most current applications (Figure 7).

During supervised learning, ART$_a$ receives a stream of patterns $\{\mathbf{a}^{(n)}\}$ and ART$_b$ receives a stream of patterns $\{\mathbf{b}^{(n)}\}$, where $\mathbf{b}^{(n)}$ is the correct prediction given $\mathbf{a}^{(n)}$. An associative learning network and an internal controller link these modules to make the ARTMAP system operate in real time. The controller creates the minimal number of ART$_a$ recognition categories, or "hidden units", needed to meet accuracy criteria. A minimax learning rule enables ARTMAP to learn quickly, efficiently, and accurately as it conjointly minimizes predictive error and maximizes code compression. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the ART$_a$ vigilance parameter ρ_a by the minimal amount needed to correct a predictive error at ART$_b$.

At the map field an ARTMAP network forms associations between categories via outstar learning and triggers search, via a *match tracking* rule, when a training set input fails to make a correct prediction. Match tracking increases the ART$_a$ vigilance parameter ρ_a in response to a predictive error at ART$_b$. A *baseline vigilance* parameter $\overline{\rho}_a$ calibrates a minimum confidence level at which ART$_a$ will accept a chosen category. Lower values of $\overline{\rho}_a$ allow larger categories to form, maximizing code compression. Initially, $\rho_a = \overline{\rho}_a$. During training, a predictive failure at ART$_b$ increases ρ_a just enough to trigger an ART$_a$ search. Match tracking sacrifices the minimum amount of compression necessary to correct the predictive error. Hypothesis testing selects a new ART category, which focuses attention on a cluster of $\mathbf{a}^{(n)}$ input features that is better able to predict $\mathbf{b}^{(n)}$. With fast learning, match tracking allows a single ARTMAP system to learn a different prediction for a rare event than for a cloud of similar frequent events in which it is embedded.

5.5 Geospatial Mapping from Satellite Remote Sensing Data

Mapping vegetation from satellite remote sensing data has been an active area of research and development over a twenty year period [Hoffer *et al.*, 1975; Strahler, Logan, and Bryant, 1978]. Recently, a new ARTMAP-based methodology for automatic mapping from Landsat Thematic Mapper (TM) and terrain data was developed to solve challenging remote sensing classification problems, using a prototype data set that predicted vegetation classification from spectral and terrain features [Carpenter, Gjaja, Gopal, and Woodcock, 1997]. After training at the pixel level, system capabilities were tested at the stand level in regions not seen during training. ARTMAP learning, being fast, stable, and scalable, overcame common limitations of back propagation, which did not give satisfactory performance on this problem. Best results were obtained using a hybrid system based on a convex combination of fuzzy ARTMAP and maximum likelihood predictions. A voting strategy improved prediction by training the system several times on different orderings of an input set. Voting also assigns confidence estimates to competing predictions.

5.6 ARTMAP-IC Applied to a Medical Prediction Problem

Automated medical diagnosis incorporates many of the most challenging problems that are intrinsic to large-scale database analysis in gen-

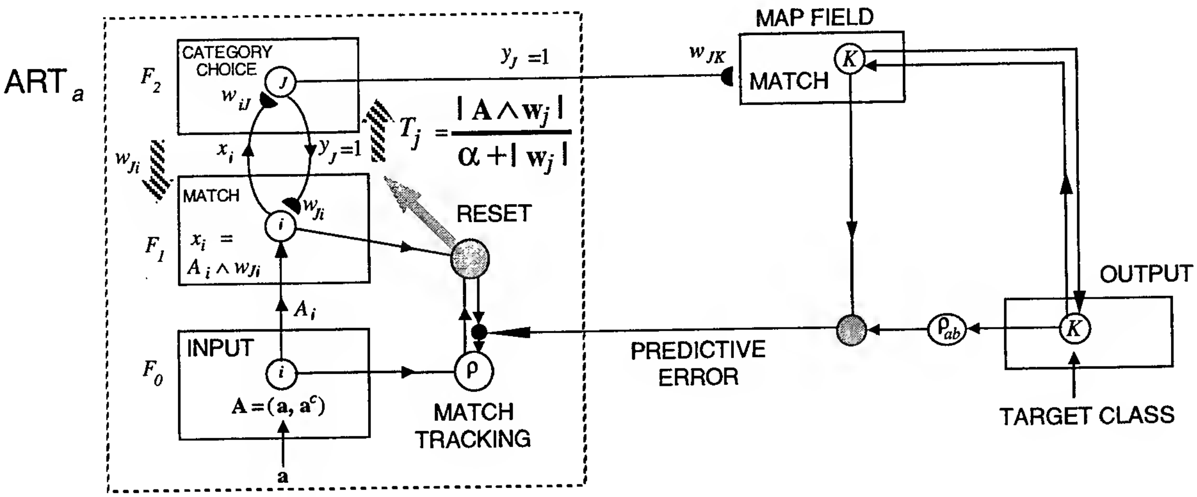


Figure 7: Fuzzy ART embedded in a simplified ARTMAP network. In the fuzzy ART algorithm, $\mathbf{w}_j$ denotes both the bottom-up weight vector and the top-down weight vector, with $w_{ij} = w_{ji}$. The ARTMAP network computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

eral. Working with these problems has stimulated a number of ART architecture developments in recent years. One such system, ART-EMAP (evidence MAP), improves performance in noisy or ambiguous input environments by adding to the basic ARTMAP system distributed spatial and temporal evidence accumulation processes, in four incremental stages [Carpenter and Ross, 1993, 1995]. Stage 1 distributes activity across category representations during performance. In a variety of studies, this device improves test-set predictive accuracy compared to ARTMAP, which is the same network except with category choice during testing as well as training. Distributed test-set category activation improves performance accuracy on various medical database simulations. Further improvement is achieved by the ARTMAP-IC neural network [Carpenter and Markuzon, 1996], which adds category instance counting and a new search algorithm to ART-EMAP distributed prediction. Instance counting weights distributed predictions according to the number of training set inputs placed in each category. A new version of the ARTMAP match tracking algorithm, which controls search following a predictive error, facilitates prediction with sparse or inconsistent data. Compared to the original match tracking rule (MT+), the new algorithm (MT-) further compresses memory without loss of accuracy. Simulations that illustrate ARTMAP-IC performance on the Pima Indian Diabetes (PID) medical database are summarized below. Results for ARTMAP-IC compare favorably to those of logistic regression, K nearest neighbor (KNN), and the perceptron network ADAP, and also compared to the basic ARTMAP network and ART-EMAP.

5.7 Comparative Simulations on the Pima Indian Diabetes Database

The PID data set [Smith, Everhart, Dickson, Knowler, and Johannes, 1988] was obtained from the UCI repository of machine learning databases [Murphy and Aha, 1992]. The database task is to predict whether a patient will develop diabetes, based on eight clinical findings: age, the diabetes pedigree function, body mass, 2-hour serum insulin, triceps skin fold thickness, diastolic blood pressure, plasma glucose concentration, and number of pregnancies. Each patient represented in the database is a female of Pima Indian heritage who is at least 21 years old. Smith *et al.* used the PID data set to evaluate the ADAPtive learning routine (ADAP) [Smith, 1962], a type of perceptron [Rosenblatt, 1958, 1962]. This study had 576 cases in the training set and 192 cases in the test set, and comparative simulations keep the same training and test sets. About 39.9% of patients in the sample developed diabetes.

Table 3 compares ADAP test set performance

Model	Correct predictions	C-index	Compression factor		
logistic regression	77%	0.84	—		
ADAP	76%	—	—		
ARTMAP (K=1) [MT+: ε=+0.0001]	66%	0.76	9.3		
	Q = 15	12≤Q≤19	Peak %	[C-index, Q]	Compression
KNN	77%	76-77%	77%	[0.80, Q=13-15]	1
ART-EMAP [MT+: ε=+0.0001]	76%	76-78%	78%	[0.87, Q=13]	9.3
ARTMAP-IC [MT+: ε=+0.0001]	79%	79-80%	80%	[0.87, Q=9-13]	9.3
	Q = 15	13≤Q≤17			
ARTMAP-IC [MT−: ε=−0.0001]	81%	80-81%	81%	[0.88, Q=15]	9.3
	Q = 11	8≤Q≤ 14			
ARTMAP-IC [MT−: ε=−0.01]	79%	78-81%	81%	[0.87, Q=9]	12.8

Table 3: Pima Indian diabetes (PID) simulations.

with that of logistic regression, KNN, and three ARTMAP networks. ARTMAP-IC uses the instance counting rule and a Q-max rule, for distributed prediction using the Q category nodes that receive maximal input. Comparative simulations show results for ART-EMAP (Stage 1), which is equivalent to ARTMAP-IC without instance counting; and for basic ARTMAP, which sets for category choice during testing. On average, the various ARTMAP networks, which share a common training regime, have 62 committed category nodes. With two output classes, an *a priori* rule-of-thumb estimate for the size of distributed category representation sets $Q = 15$. Table 3 shows that ARTMAP-IC has the best test set performance, both in terms of the C-index and the number of correct test set predictions. MT- with parameter $\epsilon = -0.01$ compresses memory even more, reducing the number of committed nodes from 62 to 45, with little deterioration in predictive accuracy. Compared to KNN, the ARTMAP networks compress memory by a factor of about 10:1.

5.8 Radar Target Recognition

Radar range profiles are one-dimensional images of radar targets that reflect the finite travel time of an electromagnetic pulse across an object [Borden, 1993; Hudson and Psaltis, 1993; Smith and Goggans, 1992]. For recognition of simulated range profiles, fuzzy ARTMAP has been shown to achieve accuracy comparable to KNN classifiers but with significantly better code compression. For automatic target recognition (ATR) from radar range profiles, ARTMAP-based methods have been developed for improved data compression and familiarity discrimination, as follows.

5.9 Reset Buffering for Code Compression

Efficient information storage is critical for applications such as missile-borne ATR where space, power, and processing speed restrictions are severe. A new fuzzy ARTMAP reset buffering technique [Grossberg, Rubin, and Streilein, 1996] promises even greater efficiency in data storage. During fast learning, the basic ARTMAP algorithm does not compute the cumulative predictive success of the categories that it learns. Slow learning and other mechanisms can incorporate some statistical factors into the trained network [Bradski and Grossberg, 1995; Carpenter, Grossberg, and Reynolds, 1995]. These techniques reduce category proliferation in response to data sets with high noise or strongly overlapping probability distributions.

ARTMAP buffering focuses on improving code compression in response to high-dimensional inputs with largely non-overlapping underlying distributions, as these are the type of distributions occurring in range profile simulations. Since it is the fuzzy ARTMAP reset process that leads to the generation of new nodes, modifying reset to be sensitive to the cumulative statistics of the training process can increase compression. Each time a node leads to predictive success, it is buffered against being reset. This operation thus uses concepts from reinforcement learning to modulate the process of recognition learning [Grossberg, 1982, 1987]. Several variants of the buffering procedure were tested on two types of radar range profile simulations. One type of simulation involved calculation of single-scatter radar returns from scattering centers in simulated aircraft. Simulations were also performed using *xpatch*, a sophisticated electromagnetic-scattering simulator developed under DoD sponsorship [Volakis, 1994].

Simulations show that buffering is capable of reducing the required storage down to its theoretical minimum with little or no loss of classification accuracy (Table 4). The MT- search algorithm of ARTMAP-IC shows similar compression results.

5.10 Familiarity Discrimination

The recognition process usually involves familiarity discrimination as well as identification. Consider, for example, a neural network designed to identify aircraft based on their radar reflections and trained on sample reflections from ten types of aircraft. After training, the network should correctly classify radar reflections belonging to these ten familiar classes, but it should also abstain from making a meaningless guess when presented with a radar reflection from a different, unfamiliar class of aircraft.

ARTMAP-FD (familiarity discrimination) is an extension of fuzzy ARTMAP that performs familiarity discrimination. During testing, an input pattern $\mathbf{A}$ is defined as familiar when a familiarity function $\phi(\mathbf{A})$ is greater than a decision threshold γ. If $\phi(\mathbf{A}) > \gamma$, ARTMAP-FD predicts an output class. If $\phi(\mathbf{A}) \leq \gamma$, $\mathbf{A}$ is regarded as belonging to an unfamiliar class and the network makes no prediction. In the case of a test set sequence, ARTMAP-FD accumulates familiarity measures at each predicted class as the sequence is presented. Once the winning class is determined, the object's familiarity is defined as the average accumulated familiarity measure of that class. The receiver operating characteristic (ROC) formalism can be used to determine the threshold γ.

Figure 8A depicts the scattering centers of a set of 36 simulated targets. The network was trained on simulated range profiles generated by 18 randomly chosen targets (in boxes), which define the set of familiar classes. ROC curves (Figure 8B) were obtained from simulated multiwavelength (40 center frequency) range profiles from all 36 targets, familiar and unfamiliar. Sequential evidence accumulation was performed for 1, 3, and 100 observations, corresponding to 0.05, 0.15, and 5.0 seconds of observation time. Classification accuracy for familiar targets was 89.5%, 97.0%, and 100.0%, with the network creating 44 category nodes.

5.11 Distributed ART and Distributed ARTMAP

Basic research continues to expand computational capabilities of ART and ARTMAP sys-

simulation	classifier	minimum # nodes	without buffering		with buffering	
			accuracy	# nodes	accuracy	# nodes
scattering centers	fuzzy ARTMAP	4	56.3%	18	55.1%	4
xpatch (SNR=3)	fuzzy ARTMAP	3	79.7%	10	77.1%	3
scattering centers	ART-EMAP	36	100.0%	135	100.0%	36

Table 4: Buffering reduces data storage requirements for simulated radar range profile ATR with minimal loss of classification accuracy.

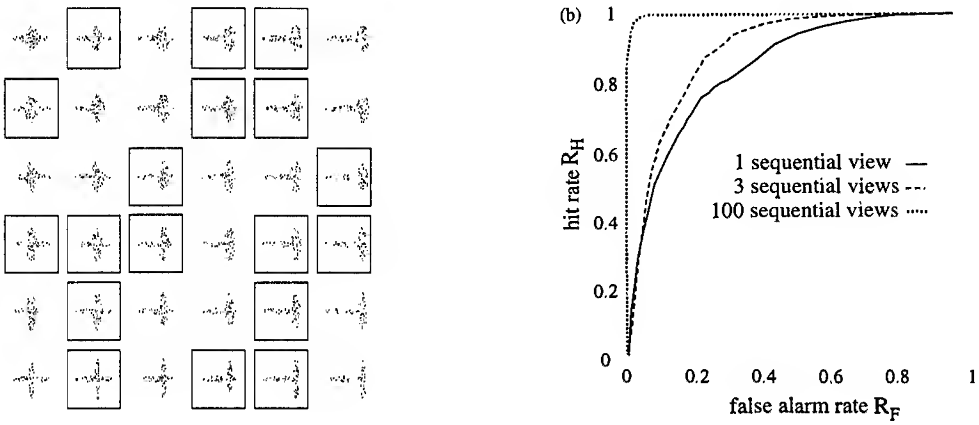


Figure 8: (A) Scattering centers for simulation of radar range profiles of familiar (boxed) and unfamiliar targets. (B) ROC curves for discrimination between familiar and unfamiliar targets using simulated radar range profiles.

tems. A new class of distributed ART models retain stable coding, recognition, and prediction, but allow arbitrarily distributed code representation during learning as well as performance [Carpenter, 1997]. These networks automatically apportion learned changes according to the degree of activation of each coding node. This permits fast as well as slow learning without catastrophic forgetting. Distributed ART models replace the traditional neural network path weight with a *dynamic weight* equal to the rectified difference between coding node activation and an adaptive threshold. The input signal T_j that activates the distributed code is a function of a *phasic component* S_j, which depends on the active input, and a *tonic component* Θ_j, which depends on prior learning but is independent of the current input, as in the fuzzy ARTMAP algorithm. At each synapse, phasic and tonic terms balance one another and exhibit dual computational properties. During learning with a constant input, phasic terms are constant while tonic terms may grow. Tonic components would then become larger for all inputs, but phasic components would become more selective, reducing the total coding signal sent by a significantly different input pattern. Dynamic weights that project to coding nodes obey a distributed instar leaning law and those that originate from coding nodes obey a distributed outstar learning law. Inputs activate distributed codes through phasic and tonic signal components with dual computational properties, and a parallel distributed match-reset-search process helps stabilize memory. When the code is winner-take-all, the unsupervised distributed ART model (dART) is computationally equivalent to fuzzy ART and the supervised distributed ARTMAP model (dARTMAP) is equivalent to fuzzy ARTMAP. With fast distributed learning, dART and dARTMAP networks are likely to expand the domain of applications of the ART family of networks.

6 Coherent Processing of Moving Targets

6.1 Motion Capture, Attention, and Tracking

When an object moves under real-world conditions, aperture ambiguity and image or detector noise often prevent all but a small subset of its image features, such as its bounding contours, from generating unambiguous motion direction cues. This problem is faced by all detection systems that use spatially limited filters, which see the world through the "aperture" defined by the filter's spatial extent. The problem is exascerbated when multiple targets are all moving in different directions in a cluttered scene under rapidly changing lighting conditions, as often happens in battlefield scenarios. Motion processing models which depend entirely on bottom-up filtering mechanisms break down badly in such situations. A process of motion capture enables the brain to cope with this problem even in response to cluttered, scintillating scenes. This project is developing a computational model of the motion capture process [Chey, Grossberg, and Mingolla, 1997a, 1997b]. Such a model facilitates the identification and attentive tracking of rapidly moving objects under noisy, cluttered, and camouflaged detection conditions.

Motion capture is the process whereby ambiguous motion signals that are distributed across an object are reorganized into a coherent representation of the object's motion direction and speed. For example, consider the task of rapidly detecting a leopard leaping from a jungle branch under a sun-dappled forest canopy. Consider how spots on the leopard's coat move as its limbs and muscles surge. Imagine how the patterns of light and shade play on the leopard's coat as it leaps through the air. These luminance and color contours move across the leopard's body in a variety of directions that do not necessarily point in the direction of the leopard's leap. Instead, the leopard's body generates a scintillating mosaic of moving contours that could easily prevent its detection. Typically, only the bounding contours of the leopard's body provide unambiguous cues of the leopard's true direction and speed of motion. Motion capture rapidly reorganizes this scintillating mosaic of moving light and shade into a coherent object percept with a unitary motion direction and speed. The leopard as a whole then seems to pop-out from the jungle background and to draw our attention.

Motion capture seems to be a preattentive, in particular, an automatic bottom-up visual process. A striking property of the model is that the same circuit mechanism which carries out motion capture can also use top-down attention to search for an object moving in a prescribed direction. This attentional priming circuit automatically suppresses motion signals that are not in the desired direction, while enhancing and grouping signals that are. Surprisingly, this turned out to be a type of Adaptive Resonance Theory (ART) circuit. As such, it suggests how the model can learn through visual experience with moving targets to group together signals that represent the same motion direction.

Figure 9 shows the model processing stages in schematic form. The model is called a Motion Boundary Contour System (mBCS) because it is homologous to the BCS that is used to form boundary representations of static forms. The static form processing system is based upon orientationally tuned computations. The motion processing system is based upon directionally tuned computations. The parts of a complex object may be defined by many oriented components, even though the object as a whole moves in a single direction. The mBCS model clarifies how signals from multiple orientations are pooled into a single direction of motion.

The model was developed through our efforts to simulate a large psychophysical and neurobiological data base about coherent motion perception. The neurobiological data concern the processing stream that joins cortical areas V1, MT, and MST. This analysis led us to articulate the following five design principles on which the model is based: (1) Unambiguous feature

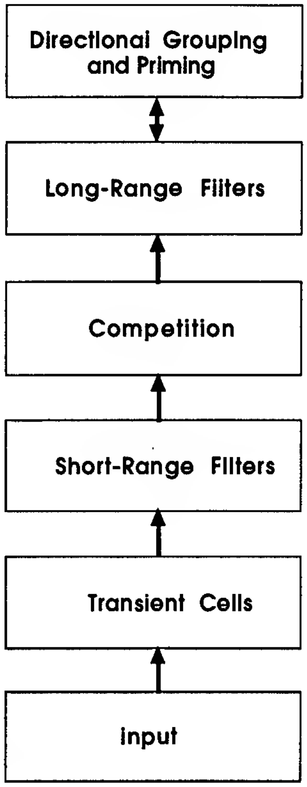


Figure 9: mBCS model processing stages.

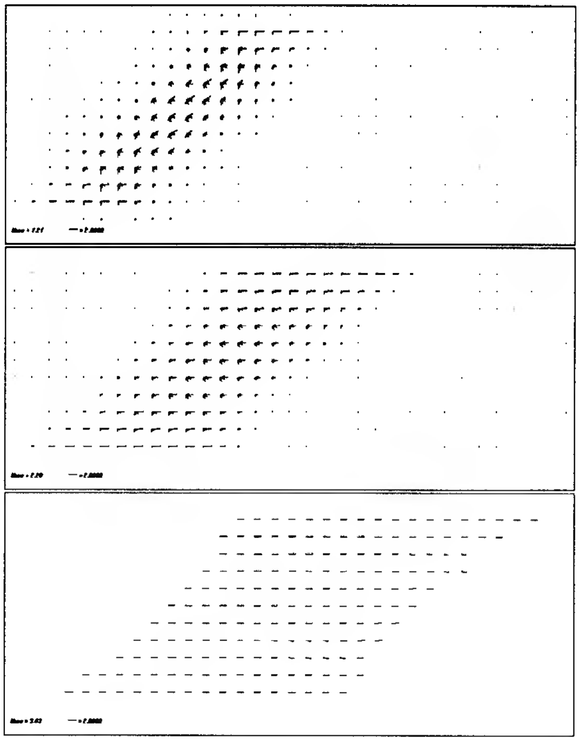


Figure 10: Motion capture develops in time. See text for details.

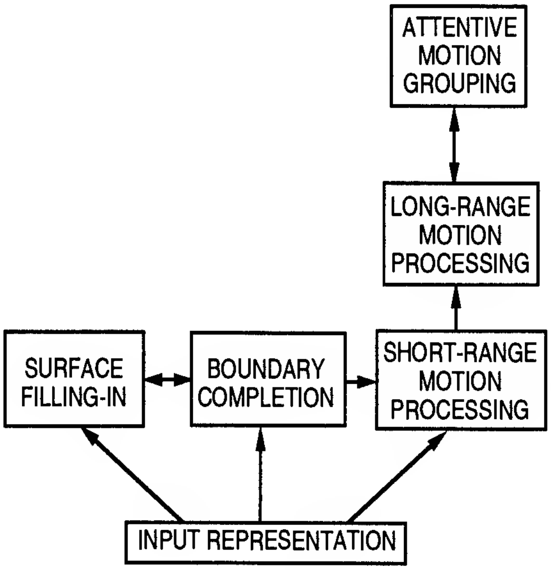

ATTENTIVE MOTION GROUPING
LONG-RANGE MOTION PROCESSING
SURFACE FILLING-IN
BOUNDARY COMPLETION
SHORT-RANGE MOTION PROCESSING
INPUT REPRESENTATION

Figure 11: Schematic of form-motion fusion model.

tracking signals are used to capture and transform ambiguous motion signals into coherent representations of object motion. (2) A single feature tracking process can contextually select both the target direction and speed. (3) Motion direction and speed are a collective property of a multiple-scale self-similar filter whose individual scales are sensitive to different speed ranges. The collective output from these scales provides robust motion estimates in response to noisy data. (4) Nearby contours of different orientation and contrast polarity that are moving in the same direction cooperate to generate a pooled motion direction signal. This process is modeled by a spatially long-range motion filter. This long-range filter also helps to solve another important problem; namely, it enables the model to track spatially and temporally discontinuous views of a target that is moving with variable speed [Grossberg, 1997]. More traditional tracking algorithms depend upon having a continuous trajectory through time. In the brain, this tracking process leads to percepts of long-range apparent motion. (5) Motion capture of the correct direction can occur without distorting the object speed estimate. This design constraint leads to the conclusion that motion capture is achieved by a spatially long-range grouping network of ART type. This grouping network operates after the long-range spatial filter. Its feedback circuit also allows attention to prime a desired motion direction.

Figure 10 illustrates motion capture for the simple case of a bar tilted at 45 degrees and moving to the right. Each successive frame shows the selected motion direction signals at a different time. Each line is proportional to the maximal activity within that time frame across all long-range filter scales that are maximally tuned to that line's direction. Due to aperture ambiguity, unambiguous feature tracking signals are at first computed only at the bar ends. The favored motion direction in the bar's middle is perpendicular to its orientation, due to aperture ambiguity. Motion capture enables the feature tracking signals, which are relatively sparse and weak compared to the ambiguous aperture signals, to suppress all aperture signals that are not compatible with them. In the bottom frame, the correct direction and speed of the bar is computed everywhere along its length.

The model is able to track both first-order and second-order motion stimuli [Baloch, Grossberg, Mingolla, and Nogueira, 1997]. A first-order stimulus is a stimulus whose motion can be discriminated by spatially tracking a difference of mean luminance or color over time. In second-order motion stimuli, there is no difference in luminance and color between moving regions, but the spatial, temporal, or ocular distribution of mean luminance or color may change through time. Illustrative second-order stimuli include scintillating objects whose mean luminance and color do not differ from that of their background. Thus the model's filtering and grouping mechanisms are designed to work under stimulus conditions that may include various types of decoys or distractors.

6.2 Fusion of Form and Motion Processing

In many biological and technological situations, an object's representation is an emergent properties of boundary and surface completion mechanisms within the form processing system. This is particularly true when sensors are used that are noisy and include many missing pixels, as also occurs within biological retinas. Under these circumstances, how can objects be tracked by the motion processing system, given that one cannot continuously track the objects' pixels from one image frame to the next? Some type of form-motion fusion is needed here to show how emergent representations of objects within the orientationally-based form processing stream interact with the directionally-based motion processing stream to generate trackable representations of objects in motion. We have been developing a computational model of how the brain accomplishes form-motion fusion [Baloch and Grossberg, 1997; Francis and Grossberg, 1996], and have used it to explain many data about how the brain knows how to combined successive image frames into motion trajectories, even when one cannot continuously track pixels from one frame to the next.

A number of authors have posited that such form-sensitive motion tracking requires pattern matching and geometry-based parsing of object representations between successive image frames; e.g., Tse, Cavanagh, and Nakayama [1997]. We have shown that the key psychophysical data about this process can be simulated simply by combining our models of boundary, surface, motion, and attentional processing into a larger architecture. In other words, form-motion fusion seems to be "nothing but" the collective action of the component models that have been reviewed above. A schematic of this architecture is shown in Figure 11.

At what processing stage should the link between form and motion processing be made? It should occur after the form processing stage at which emergent binocular boundaries are generated and before the long-range motion filter at which motion signals are pooled into motion directions; see Figure 11. This interaction also helps to generate accurate representations of a moving object's depth in cases where range detectors are not used. This is true because the form processing stream uses precisely oriented binocular matches to generate precise representations of a static form's depth. The motion processing system loses this capability by pooling multiple orientations into a single motion direction. The form-motion interaction selectively enhances those motion computations that are consistent with the depths that are computed within the form system. In the visual cortex, this larger architecture models how the form processing stream between cortical areas V1, V2, and V4 interacts with the motion processing stream between cortical areas V1, MT, and MST via a cross-stream interaction from area V1 to MT.

7 Psychophysical Experiments for Real-World Tasks

7.1 Visual Search Experiments in Cluttered Environments

Much effort has gone into developing algorithms for helping human observers on the battlefield to make more effective decisions. Surprisingly little work has been done, however, to determine the types of perceptual and cognitive factors that can improve the accuracy of human decision makers under naturalistic conditions. Many decision making tasks involve rapid search through cluttered and noisy sources of visual information. Many visual search experiments published in the cognitive science literature use small numbers of isolated targets and distractors with simple, 2-D forms. On the other hand, much of the human factors research on more complex imagery is so tied to measures of performance on specific image paradigms that an elucidation of fundamental limits of human mechanisms is difficult to achieve.

To more closely approximate natural scenes than is typically done in psychological research, we displayed on a video monitor variable-sized rocks in formations that resembled irregular stone walls, rendered using a 3D modeling program. We examined visual search for a foreground rock resting on other rocks. Experimental factors were (1) cast shadows (present or absent) and (2) occlusion configuration (amount of contour occluding other rocks and whether or not any background rock was visible on both sides of an occluding target rock). Typical scenes contained between 20–160 rocks. In a "target present" scene, one rock was closer to the observer and in front of the others; in a target absent scene, all rocks were approximately the same distance from the observer. Illumination was modeled so as to ensure soft shadows on shadow-present scenes by a pair of point light sources: a faint source in line with the viewing direction, and a bright source from above and slightly to the right of the line of sight.

A psychophysical display paradigm for parametrically varying complex variables of illumination and shading in a visual search task has been developed. Certain published findings based on search in scenes of simple polyhedral objects, including asymmetries in search speed for illumination from above vs. below, have been shown not to generalize to complex, smoothly-shaded images. Additional results suggest useful ways of manipulating highlight and shadow information in rendered imagery to encourage the "pop-out" of targets in clutter. A preliminary report appears in Cunningham *et al.* [1996].

7.2 Experiments on the Perceived Segregation of Element-Arrangement Textures

Recent progress in the generation of color-fused imagery from artificial sensors, operating in the visibile-near-infrared (low-light CCD) and thermal infrared bands (Gen III intensifier tubes), indicates that appropriate combinations of wavelengths used to render imagery may make an operator's task of detecting targets in clutter significantly easier than would be the case viewing gray-scale imagery [Waxman *et al.*, 1997]. The fusion algorithms developed by Waxman's group are based in part on a foundation of modeling and algorithm development by our own group. Development of these models and algorithms, in turn have been based on, among other constraints, analysis of human psychophysical experiments conduct by our own group and by others. Thus, a cycle of experiment measurement, theoretical modeling and simulation, followed by development of applications can be closed to further advance all three endeavors. Additional psychophysical work, described next, is being targeted to refine our understanding of human mechanisms for segmentation of chromatic images, which will contribute to the development of still more sophisticated algorithms and hardware for color-fused imagery.

The fundamental visual mechanisms involved in such visual tasks are as yet poorly understood, however, and quantitative experiments in chromatic texture segregation ought to pay direct dividends in helping to select parameters for rendering of color-fused imagery.

Many experiments have used achromatic element-arrangement patterns to explore the mechanisms involved in texture segregation. Element-arrangement patterns are composed of two types of elements arranged in alternating vertical stripes in the top and bottom regions and in a checkerboard pattern in the middle region. Only recently has work been done with chromatic element-arrangement patterns and early results suggest that different mechanisms are involved. We undertook a series of experiments to determine the factors that are primarily responsible for the perceived segregation of chromatic element-arrangement patterns. We investigated the effects of hue, spatial scale, and background luminance on the segregation of the element-arrangement patterns. Hue similarity, as rated by subjects in a separate procedure, was a relatively weak factor for predicting perceived segregation. The effects of brightness differences and luminance differences interacted with background luminance and spatial scale. Perceived segregation was stronger with a black background than with a white background and stronger for higher spatial frequencies.

An equation based on differences in color-opponent channels has been developed that accurately predicts the perceived segregation of chromatic element-arrangement patterns. A theory that accounts for the background luminance affecting chromatic and achromatic patterns differently has been developed and supported by experimental findings. The experimental results suggest that perceived segregation of chromatic element-arrangement patterns is largely a function of cone contrast and perceived segregation of achromatic element-arrangement patterns is largely a function of the contrast ratio of the squares.

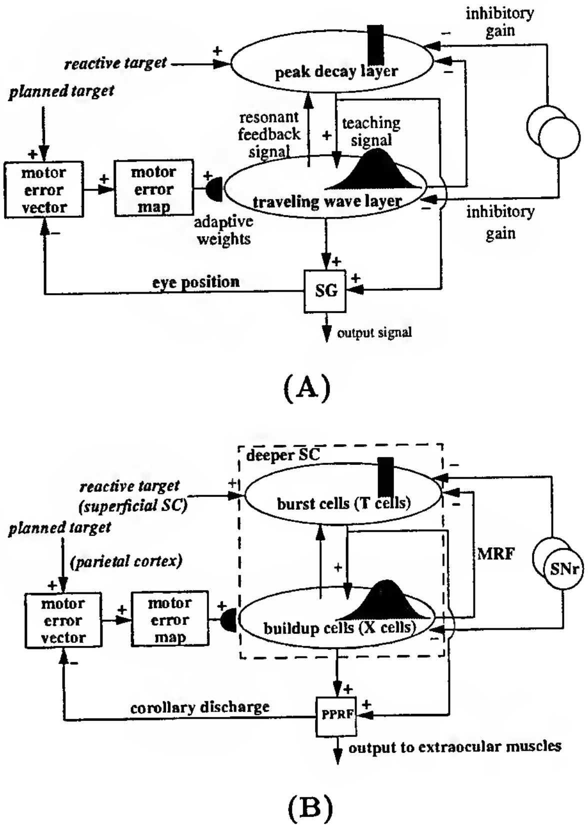


Figure 12: (A) Model of saccadic learning and movement control. (B) Anatomical correlates: SC=superior colliculus; SNr=substantia nigra, PPRF=paramedian pontine reticular formation, MRF=mesencephalic reticular formation.

8 Adaptive Multimodal Fusion of Eye Movement Commands

This project investigated how multiple sources of information that are computed in different coordinate frames can learn in real-time to fuse their information into a common movement map. All the movement constraints can then compete for attention on the map and select a movement target. Such an algorithm is useful in tracking systems wherein changes in sensor properties due to use in the field may require a self-calibrated adjustment of decision-making parameters in order to maintain accurate target tracking. The neurobiological insights about how the brain allocates attention to movement targets may also help to design more effective displays for pilots and other users of visually formatted information.

The project modeled how this type of adaptive fusion controls the rapid ballistic movements that our eyes make when we look around the world [Grossberg, Roberts, Aguilar, and Bullock, 1997]. These ballistic movements are called saccades. We modeled how the saccadic movement system selects a target when visual, auditory, and planned movement commands differ. In particular, visual signals are computed in retinal, or camera-centered, coordinates, whereas auditory signals are computed in head-centered coordinates. Much evidence suggests that saccadic commands are computed in motor error coordinates. How do all these coordinate systems get consistently calibrated through learning, and how do they interact to select a movement command from all the targets that may be available at any time?

Recent neurobiological data suggest that this sort of multimodal data fusion takes place in the deep layers of the superior colliculus (SC). The model suggests how auditory, and planned saccadic target positions become aligned and compete with visually detected target positions to select a movement command (Figure 12). For this to occur, visual targets are transformed from retinotopic to motor error coordinates, and a transformation between auditory and planned head-centered representations and this motor error representation is learned. The model simulates recent neurophysiological data recorded from identified cells within the deep layers of cat and monkey SC. These cells are of great functional interest because one cell type (the peak decay, or burst cell) is predicted to be a source of movement and learning signals before the multimodal map gets learned. The other cell type (the traveling wave, or buildup cell) is proposed to receive the peak decay teaching signals, as well as the multimodal auditory and planned movement signals with which they are associated in the movement map. In fact, NMDA receptors, which are known to be involved in brain learning, have recently been discovered in this part of the SC. After the map is learned, both cell layers use same circuit that controls the learning process to focus attention and select a movement target. The model also functionally interprets the saccade-related behaviors of cells in many of the areas that interact with SC in making a movement decision, including the frontal eye fields, parietal cortex, messencephalic reticular formation, paramedian pontine reticular formation, and substantia nigra pars reticulata (see Figure 12).

9 Head Mounted Space-Variant Active Vision System: Algorithms and Hardware

The goal of this project is to construct a miniature space-variant active vision system which is to be demonstrated as a prosthetic device for the blind. Other related applications of this technology to head mounted miniature active vision systems, for example using infra-red sensors, night vision applications, etc. are expected to also benefit from this technology. The benchmark tasks for the computer vision system are to perform a variety of pattern recognition, navigation, and obstacle avoidance tasks. The man-machine interface will be provided via auditory cues in the form of virtual (i.e., spatially defined) audio cues, and other auditory stimuli.

The motivation for this project is to harness the large potential reduction in computational complexity provided by the architecture of the higher vertebrate and human visual systems, which are in all cases strongly space-variant. Previous work has shown that it is possible to achieve two-four orders of magnitude in reduction in space-complexity in unconstrained wide-field machine vision applications via the use of space-variant architectures such as the log-polar image format.

Space-variant active vision systems are ideal as an architectural basis for the construction of miniature computer vision systems, but a number of difficult problems are associated with exploiting this feature of biological vision. These include the need to build miniature camera systems [Engel *et al.*, 1994], develop control algorithms [Greve, Engel, and Schwartz, 1997], and most significantly, to develop the basic image processing algorithms that are applicable in this domain.

In the first year of this project, our goal was to construct a hardware system suitable for prototyping the algorithms for the project, and to begin design of the algorithms. This initial hardware system has been built from "off-the-shelf" parts, and provides a much larger degree of computational support than is envisioned for the final embedded version of this system. This extra functionality (large hard disk, Pentium PC development environment, large memory, etc.) is intended to facilitate the early development stages of the project. The platform is based on the use of a quad SHARC digital signal processing system, providing 160 MFLOPS on four Analog Devices 2106x processors, hosted by a Pentium PC, and deploying a miniature space-variant active vision system.

The principal results of the past year have been two significant advances in basic algorithms for space-variant active vision. One of the most serious blocks to the development of computer vision systems based on space-variant active vision has been the lack of a general image processing toolkit for the difficult image format provided by the log-polar (and related) space-variant sensor formats. During the past year, we have completed development of an "exponential chirp algorithm" which provides the equivalent of the 2-D Fourier Transform (FFT), but which works on the log-polar image format. The exponential chirp provides a form of "quasi-shift invariance", and we have demonstrated its successful use in pattern matching tasks. Of significance is the fact the we have developed a fast exponential chirp algorithm which has identical complexity as the conventional 2-D FFT. Since the principal motivation for using log-polar image formats is that they provide image sizes which are two to four orders of magnitude smaller than the corresponding constant resolution image input, there is a large improvement in processing speed. Since the exponential chirp can be used, as is the case for the FFT, for all aspects of image processing from early vision (e.g. filtering) to "late" vision (e.g. pattern recognition), we believe that this represents a fundamental advance in this area. We have demonstrated a 30 frame/sec processing rate (on a 180 MHz Pentium P-6) using these methods.

A second major result has been the development of fast "anisotropic diffusion" methods for image segmentation. Image enhancement and segmentation methods based on the implementation of partial differential equation based "nonlinear" diffusion methods provide significantly better edge enhancement and segmentation than simple Laplacian or isotropic based diffusion techniques. The problem with exploiting these superior results in machine vision systems has been the extremely large computational load created by solving nonlinear partial differential equations (PDEs) on each image. We have found, for example, that on a Pentium P-6, it can take up to 2 minutes to process a single image frame with an anisotropic diffusion method. By combining our "fast" methods for anisotropic diffusion with space-variant image architecture, we have achieved 30 frame/sec real time processing rates which are indistinguishable in quality from the conventional nonlinear diffusion results. Our original work on nonlinear diffusion is described in a recent series of papers [Fischl, 1996; Fischl, Cohen, and Schwartz, 1997a, 1997b; Fischl and Schwartz, 1996, 1997a, 1997b, 1997c], while the exponential chirp algorithm is described in Bonmassar and Schwartz [1996a, 1996b, 1996c]. In this section, we will briefly outline some of these results.

9.1 Real Time Adaptive Alternatives to Nonlinear Diffusion in Image Enhancement

Many early vision systems employ some type of filtering in order to reduce noise and/or enhance contrast in regions which correspond to borders between different objects within an image. The logical extreme of this process is the creation of a piecewise constant image with step discontinuities at region boundaries. This goal is unattainable using linear filtering techniques, as noise reduction blurs the locations of boundaries between regions, sometimes to the point of fusing them. In order to address this problem, Perona and Malik [1990] introduced a nonlinear version of the diffusion equation previously used by Koenderink and Hummel [Hummel, 1986; Koenderink, 1984] for early visual processing. In this formulation, image intensity is treated as a conserved quantity and allowed to diffuse over time, with the amount of diffusion at a point being inversely related to the magnitude of the intensity gradient at that location. This process produces visually impressive results in terms of the creation of sharp boundaries separating uniform regions within an image, but is computationally expensive (see ter Haar Romeny [1994] or Fischl and Schwartz [1997a] for a more complete discussion of these issues). Linear diffusion is identified with Gaussian filtering because the Gaussian is the Green's Function of the linear diffusion equation for an infinite domain. Thus, spatial integration with Gaussian kernels can be used to implement linear diffusion. The nonlinear anisotropic diffusion proposed by Perona and Malik has no known closed form solution analogous to the Green's Function solution of the linear equation, and therefore must be integrated numerically.

$$I_t = \nabla \cdot (c(\mid \nabla I \mid)\nabla I) \tag{1}$$

where I is the intensity image, c is a diffusion coefficient, I_t is the partial derivative of I with respect to time, and Δ is the Laplacian operator with respect to the spatial coordinates. The solution to equation (1) can be written in terms of the Green's function[1] of the system as

$$I(x,y,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x',y',0)G(x,x',y,y',t)dx'dy' \tag{2}$$

where $I(x',y',0)$ is the initial image, and the Green's function $G(x,x',y,y',t)$ is a Gaussian kernel given by

$$G(x,x',y,y',t) = \frac{1}{\sqrt{4\pi ct}}e^{-\frac{(x-x')^2+(y-y')^2}{4ct}} \tag{3}$$

The Green's function $G(x,x',y,y',t)$ is the kernel of the integral operator which is the inverse of the diffusion operator. Thus, convolution with larger scale Gaussian kernels is equivalent to the evolution of the diffusion equation on an infinite domain for longer periods of time, with the original image as initial conditions:

$$I(x,y,t) = \int\int_D I(x',y',0)G_t(x,x',y,y',\nabla I)dx'dy' \tag{4}$$

where D is the image domain and we subscript the kernel function $G_t(x,x',y,y',\nabla I)$ with t to emphasize the fact that different kernel functions exist for different evolution times.

9.2 Anisotropic Diffusion and Nonlinear Filtering

The nonlinear diffusion equation yields impressive image enhancement by selectively averaging intensity values from one side of an edge or the other, but not both. This can be seen be tracking the path through which intensity values diffuse as the equation is integrated, then viewing them as kernels. The spatial integration of the resulting kernels with the initial image will *exactly* mirror the numerical integration of the diffusion PDE. Carrying this procedure out effectively lets one view the nonlinear filter enacted by the integration of the diffusion equation.

In order to build the equivalent filter we must first specify a numerical implementation of (1). We use a simple scheme derived in Fischl and Schwartz [1997a]. Given the initial image at time t_0, the image at time $t_0 + \Delta t$ can be generated by correlating the initial image with a set of space and time varying masks:

$$I(x,y,t_0{+}\Delta t) \approx \sum_{x'}\sum_{y'} K^{t_0}_{x,y}(x',y')I(x{+}x',y{+}y',t_0) \tag{5}$$

where the mask weights are given by

$$K^{t_0}_{x,y} = \frac{\Delta t}{2}\begin{bmatrix} 0 & c^N(t_0) & 0 \\ c^W(t_0) & \frac{2}{\Delta t} - (\sum_{i \neq 0} c^i(t_0)) & c^E(t_0) \\ 0 & c^S(t_0) & 0 \end{bmatrix} \tag{6}$$

In order to construct the diffusion kernels $C^t_x(i)$ which parallelize the diffusion, we proceed inductively. For each point x in the image, we create a kernel C_x and initialize it using a Kronecker delta function

$$C^0_x(i) = \delta_i = \begin{cases} 1 & i = 0 \\ 0 & i \neq 0 \end{cases} \tag{7}$$

[1] More accurately, the Green's function is the Gaussian multiplied by a temporal step function [Barton, 1989].

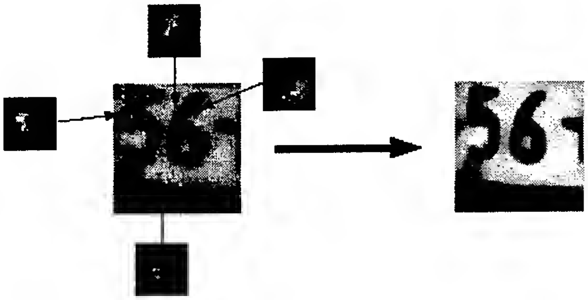


Figure 13: Original image at left shown together with the diffusion kernels for the four indicated locations. The image at the right was generated by spatially integrating the full set of kernels with the initial image.

$$C_x^{t+1}(n) = \sum_i K_x^t(i) C_{x+i}^t(n-i) \qquad (8)$$

The results of carrying out this procedure on the mean curvature-based diffusion process of El-Fallah and Ford [1994] are shown in Figure 13.

9.3 Results

A summary of the results obtained by these approaches is shown in Figure 14. On the left is shown the original figure, with a Canny edge operator applied to the bottom left frame. (The choice of the edge operator is simply to allow a visualization of the quality of the edge enhancement). The full anisotropic diffusion method produces the benchmark figure shown second from the left, with a processing time of roughly two minutes on a Pentium P-6 (180 MHz). Third from the left is shown our Green's Function Approximator [Fischl and Schwartz, 1997a], with a processing time of roughly 10 seconds on the same processor (i.e., 10 fold increase in speed). Finally, in the last frame is shown the offset filter approach to nonlinear diffusion [Fischl and Schwartz, 1997b], with a speed increase of 50 relative to the anisotropic diffusion, and a processing time of roughly 2 seconds on the same processor.

In summary, we have achieve a 50 fold increase in processing time, while retaining the favorable image appearance of nonlinear diffusion. The final step towards real-time performance is then provided by applying these methods to a space-variant image frame, yielding a full 30 frame/sec processing rate [Fischl, Cohen, and Schwartz, 1997b].

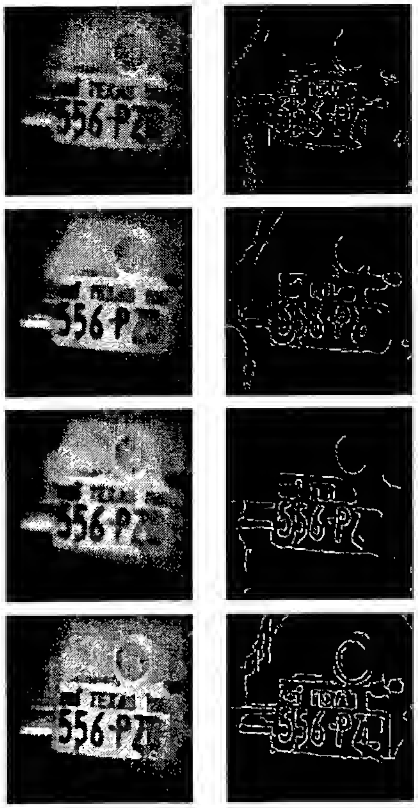


Figure 14: From left to right: original, anisotropic diffusion, gfa, offset median.

The final section of this summary will briefly review the exponential chirp algorithm. This work is described in detail in Bonmassar and Schwartz [1996a, 1996b, 1996c].

Figure 15 shows an example of an image (the eye of Lenna), in log-polar image format. On the left is shown the original image pair (i.e., the image, on the bottom, and its log-polar transform on top). On the right is shown the reconstruction of these image pairs obtained by applying the exponential chirp transform, and then applying the inverse exponential chirp transform. The forward, followed by inverse, exponential chirp transform is expected to produce the identity, if the difficult space-variant image sampling and are correctly solved. This is the case, as the original log-polar and retinal image is virtually indistinguishable from from the final result after passage through forward and inverse exponen-

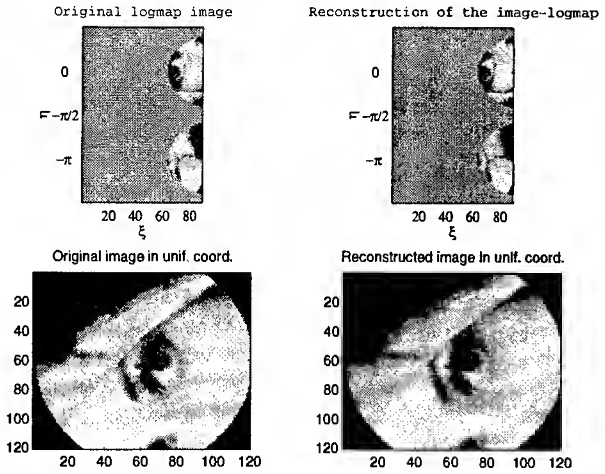


Figure 15: Exponential chirp reconstruction.

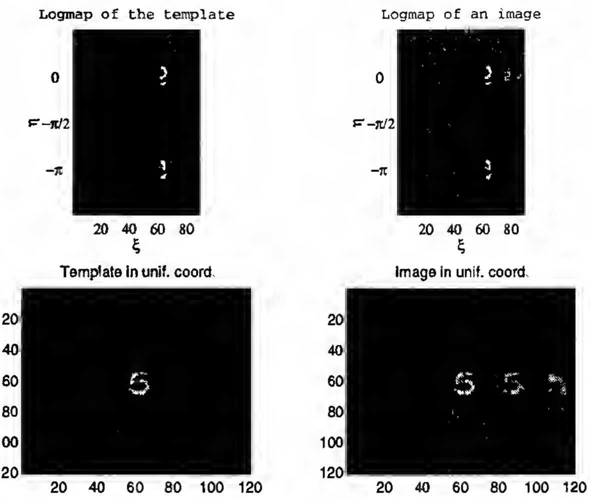


Figure 16: Pattern matching with exponential chirp.

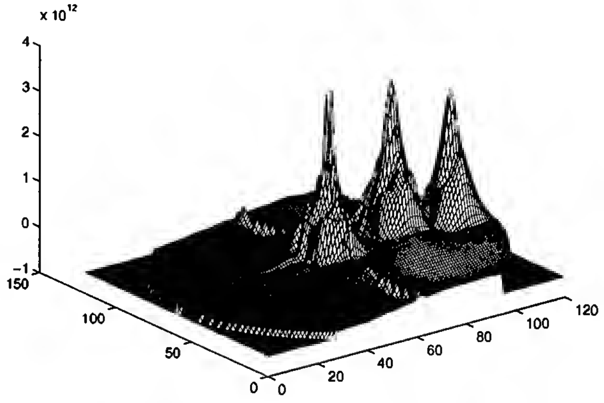


Figure 17: Correlation peaks with exponential chirp.

tial chirp.

Figure 16 demonstrates the chief image processing problem in space-variant vision, which motivated the development of the exponential chirp transform. On the top left is shown a centrally fixated numeral "5", and the left is shown three shifted versions of the "5". As expected from the properties of space-variant sampling, the bottom left frame shows the degradation in resolution due to shifting the object away from the center of gaze. On top is shown the corresponding log-polar representations of these image frames. On the right, it can be seen that the log-polar representation of the shifting "5" in the image plane changes both in size and shape. This aspect of space-variant imaging is one of the chief algorithmic problems in the field, since even simple pattern matching operations are significantly complicated by the space-variant geometry of the sensor.

Figure 17 shows the application of a correlation operator, based on the application of the exponential chirp algorithm, It can be seen that the target ("5" in this case) is matched with good signal to noise at all positions in the log-polar plane. This matching operation is performed, via the "fast" exponential chirp algorithm, at 30 frames/second on a Pentium P-6 processor [Bonmassar and Schwartz, 1996c].

In summary, by combining fast anisotropic nonlinear diffusion methods with the fast exponential chirp transform, it is possible to achieve frame rate processing (30 frames/second) on current generation (e.g. Pentium P-6) processors. The demonstration of these algorithms, implemented on a miniature space-variant active vision system, will take place within the current year.

10 Robotic Navigation under Visual Guidance

This project applies neural network models to the tasks of visual recognition and navigation using mobile robots. The long-term goal is to utilize mobile robots as platforms with which to test and refine neural network models developed in the department of Cognitive and Neural

Systems. The work emphasizes self-organizing, unsupervised architectures that can ultimately afford greater robustness and flexibility than traditional approaches. This section describes some preliminary results.

10.1 Extracting Depth Information without Camera Calibration

As a first step, P. Gaudiano and E. Sahin have developed a method for real-time object localization from monocular camera motion, and tested it using a mobile robot equipped with an on-board camera and image processing system. The goal is to integrate this method with an unsupervised neural controller that learns the forward and inverse odometry of a differential-drive mobile robot [Gaudiano, Zalama, López-Coronado, 1996]. The unsupervised controller needs to obtain information about the position of targets in the environment: the algorithm described below relies on movement and does not require camera calibration.

Localization of an object within an egocentric coordinate frame is computed using the *looming effect*, which relates the change in the size of an object to the change in the camera position when the camera is moving perpendicular to the object's plane. The looming method is usually preferable to optical flow methods because of its robustness, computational simplicity and independence of camera calibration.

The angle between the object and the robot's direction is also easily computed, in this case using the horizontal position of the object on the image. Although the quantitative computation of this angle requires camera calibration, the uncalibrated information is useful for the unsupervised neural controller.

The looming method has been implemented on the Pioneer 1 (Figure 18A) by Real World Interface, a two-wheel differential-drive mobile robot, equipped with a color CCD camera and a Cognachrome 2000 color vision system by Newton Research Labs. This vision system can identify and track "blobs" of user-selected color at a 30Hz frame rate. Specifically, the vision system returns the coordinates of the centroid, width, and height of the blobs in the image. For the looming method, these quantities are tracked as the robot moves directly toward or away from the object.

When the robot's displacements are known through internal odometry, the object's distance can be derived by comparing as few as two frames, which on our system are processed at a rate of 10Hz, except if the object is very far or if the robot moves slowly. Then more frames may be required. Figure 18B illustrates typical performance: the robot moves back and forth at a velocity of 15mm/sec, starting at a distance of about 1,000mm from an object. Once the robot has moved about 50mm, the distance estimate is already within about 20mm of the actual distance. One goal is to combine this technique with the Gaudiano *et al.* [1996] model of unsupervised robot control so that the robot's odometry can be learned, rather than having to be obtained through external calibration. Another goal is to cast the same algorithm in the form of an adaptive neural network, whereby learning of the relationship between looming and depth can be carried out in a totally unsupervised, autonomous fashion.

10.2 A Neural Model of Attentive Vision

P. Gaudiano and A. Harner are developing a neural network model of attentive visual search, whereby to speed the selective processing of complex scenes. The project adapts the Spatial Object Search (SOS) model of Grossberg, Mingolla, and Ross [1994]. It consists of two stages: a pre-attentive processing stage that operates locally and in parallel over a low-resolution image of the visual field, followed by attentive processing that operates on small, high-resolution windows in series.

The first stage forms a number of low resolution feature maps by convolving the image with a set of feature filters. Because the model aims at machine vision applications, the model uses a fast oriented pyramid. The low-resolution feature maps are then gated by a feature vector representing the total amount of each feature

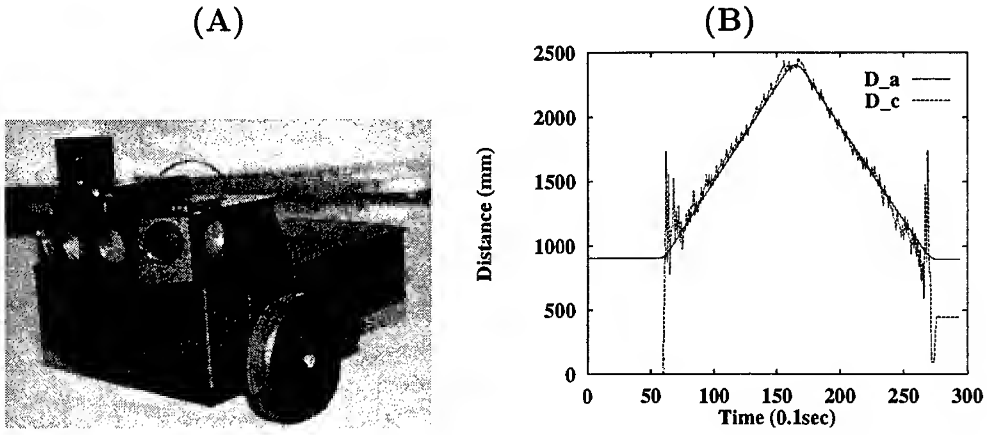


Figure 18: (A): The Pioneer 1 mobile robot. (B): Real distance (solid) and distance estimated with the looming method (dashed).

stored in long-term memory for a given target. The gated feature maps are then combined to form a composite gated feature map. After this map is smoothed, the peaks on this map mark regions of interest to be processed in more detail by the second stage of the model. The second stage positions a high-resolution, mobile window over these peaks. This window captures the role of attention in selecting and positioning the fovea over areas of interest. A template match is then performed on images within the high-resolution window using a network similar to ART2a, which learns to recognize the target features to be recognized.

The model is robust and fast when working with large noisy images. It is able to find targets consisting of a conjunction of several features in a scene with multiple distractors sharing some of those features. When a significant amount of noise is added to the system, it still finds the target (as shown in Figure 19(A)). The model's response time resembles the reaction time (versus number of distractors) curves seen in psychophysical experiments. When the target and distractors do not share many features, it mimics the "pop-out" effect by finding the target quickly and independently of the number of distractors. In addition, the model is able to find and recognize textured objects in textured backgrounds to a limited degree without performing a pre-attentive segmentation process (as shown in Figure 19(B)). The system takes only about 8 seconds to train and 10 seconds to test on 256 × 256 images in MATLab running on a Sun Sparc 10 workstation.

11 Neuromorphic VLSI for Battle Awareness

Despite many years of research and remarkable advances in the field, the problem of robust automatic object recognition by *truly autonomous, highly mobile and portable hardware systems*, is still an open problem [U.S. Army Research Office, 1994]. Biological organisms excel at solving problems in sensory communication—audition and vision—and motor control, by sustaining high computational throughput with minimal energy consumption and heat production. Biological systems are highly mobile and thus constrained by size, weight, and the availability of energy resources. Thus it may be worthwhile to implement biological information processing principles in VLSI designs.

Here we describe our level of effort in developing sensory processing hardware for automated sensing and vision systems at the Boston University VLSI and Neural Net System Laboratory in collaboration with Johns Hopkins' Sensory Communication and Analog VLSI Laboratories. More specifically, we are developing real-time computational engines for image preprocessing, learning, and adaptive classification

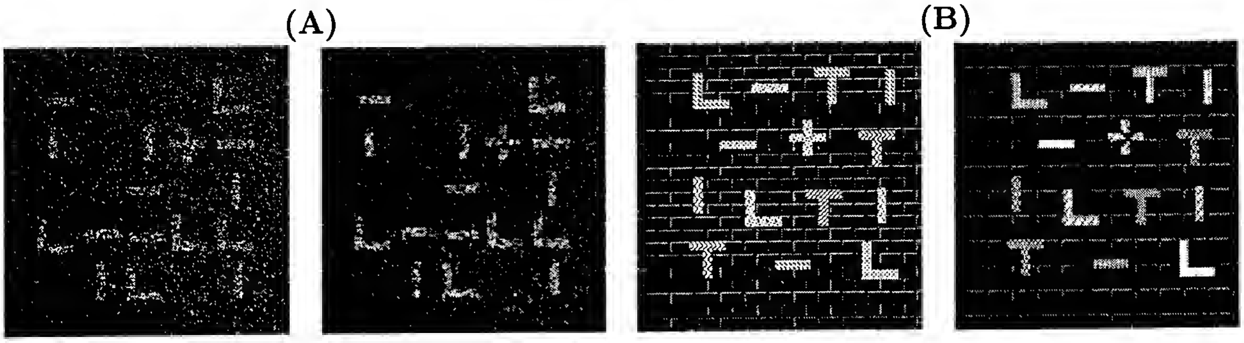


Figure 19: (A) Left: A test image with 300% noise. Right: The system finds the target (the white cross) in one try and marks it with a black 'X'. (B) Left: A textured test image. Right: Again, the system finds the target.

based on the BCS/FCS architecture and the ART family of learning networks, respectively. As a demonstration vehicle and case study problem we focus on a real-time visual processing system to enhance Synthetic Aperture Radar (SAR) sensor data for viewing by human observers. The processed SAR image will be received in a buffer for use by Intelligent Analysts or automatic machine classifiers.

Our work bridges three clearly established disciplines: (1) neural and cognitive science, (2) VLSI signal processing, and (3) computer architecture. Engineers doing work on computer architecture don't know much about neural systems. Similarly researchers in computational neuroscience don't know much about VLSI signal processing and computer architecture. Most computer architecture work is done today in companies driven by commodity markets and not by highly efficient systems for specialized problems; this is likely to change as the availability of design tools for custom and semi-custom integrated circuit design have reached maturity. The work reported here is orthogonal to on-going VLSI signal processing and architecture for video processing using specialized digital signal processor architectures (VSPs).

In the remainder of this section we discuss in detail our work on VLSI architectures for the BCS/FCS and for the ART family of learning systems.

11.1 VLSI Architectures for the BCS/FCS Model

Work in this subtask of the project has focused on the following basic questions: (1) Can we simplify mathematical functions of the model to map on parallel analog hardware (resistive grids) and how do these simplifications affect performance? (2) What is the appropriate signal representation and coding at each stage of the model and how do we solve the global interconnect problem?

The original BCS/FCS model is a multi-stage, multi-scale visual processing system depicted in Figure 20. Currently, our simulations have shown that a reduced BCS/FCS model using only stages 1, 2, 3 and 9 would speed prototype development while yielding good results. It should be emphasized that when we refer to "simplifications" throughout the discussion that follows these are done after extensive simulations and analysis of the original models and more often than not, simplifications are used as stepping stones for more complicated model development.

11.2 Shunting Neuron and Resistive Grids

The first question that has been addressed and reported in Hinck and Hubbard [1996] relates to the implementation of shunting neurons and the Gaussian kernels. Resistive grids [Andreou *et al.*, 1995; Boahen and Andreou, 1992; Mead, 1989] are powerful computational prim-

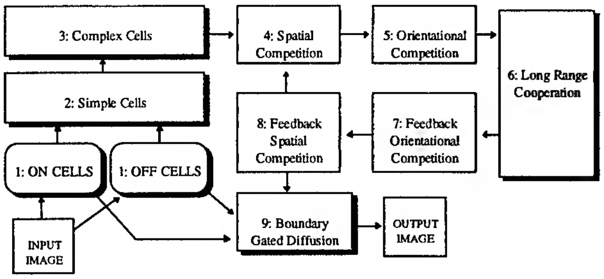

3: Complex Cells
2: Simple Cells
1: ON CELLS
1: OFF CELLS
INPUT IMAGE
4: Spatial Competition
5: Orientational Competition
6: Long Range Cooperation
8: Feedback Spatial Competition
7: Feedback Orientational Competition
9: Boundary Gated Diffusion
OUTPUT IMAGE

Figure 20: Block diagram of a single-scale BCS/FCS model. Stage 2 through Stage 8 are represented by single blocks except each contain K sub-blocks. These stages contain spatial filters that are rotated by an angle determined by the number of K orientation. The prototype will contain 4 orientation initially, although 8 orientations are anticipated as the prototype is revised.

itives that can yield Gaussian like kernels. We address these issues by focusing on the the first stage in the BCS/FCS architecture.

Our "shunting neuron" circuitry is based on the Hodgkin-Huxley cell electrical equivalence model. The model is composed of three branches: a resting potential, a excitatory and inhibitory branch. In the original model, the excitatory and inhibitory branches were variable conductances whose magnitude it computer using a Gaussian convolution operator. The variable conductances were implemented using modified four-quadrant voltage-to-current multipliers where one input was used as a gain factor and the other as a differential voltage.

The final shunting neuron was realized by eliminating the resting potential branch and transforming into a pull-up pull-down structure. Our simulation results from an array of "shunting neurons" provides evidence that our analog resistive grid based circuitry works as a close approximation to the original BCS/FCS specification.

The comparison between Gaussian kernels and resistive grid based kernels was done by simulating Stage 1 of the BCS/FCS architecture. A wide dynamic range input function, as depicted in Figure 21 to test the circuit's dynamic compression and edge enhancement abilities. As a benchmark, we build two mathematical MATLAB model based on the original BCS/FCS with either a Gaussian or Exponential filter (resistive grid based kernel).

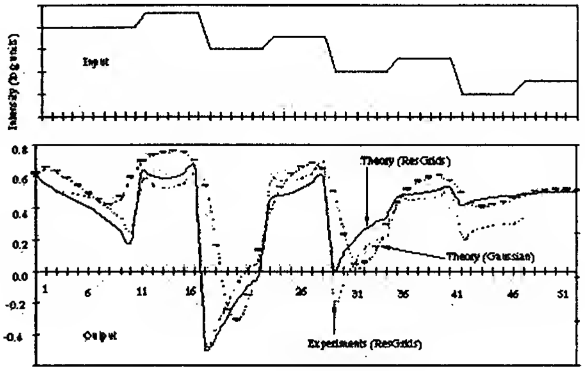


Figure 21: Comparison between Gaussian kernels and resistive grid based processing.

Comparing the BCS/FCS (Exponential, MATLAB) model output with our circuit output shows they are the same for larger inputs and slightly different for smaller inputs. The difference for smaller inputs can be attributed to have different parameter values, because some of the circuit parameters value were set inherently by the circuit themselves. In comparing the BCS/FCS (Gaussian, MATLAB) model with the other two models, the Gaussian look much smoother. Besides from comparing the circuit's output to the other models, we investigated the circuit's ability to compress a large dynamic range and provide edge enhancement. The circuit can compressed four orders of magnitude of input intensity, normalized the results between 1 volt and provide edge enhancement over all four orders of magnitude.

11.3 Simplified BCS VLSI Architecture

We have explored implementation issues through the simplified architecture, shown schematically in Figure 22, which partitions the BCS model into three levels: simple cells, complex and hypercomplex cells, and bipole cells [Waskiewicz and Cauwenberghs, 1997]. Complex cells compute unidirectional gradients of normalized intensity obtained from the

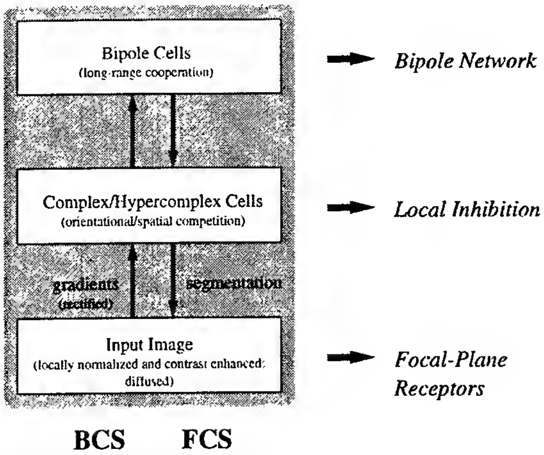


Figure 22: BCS/FCS model of image segmentation, feature filling, and surface reconstruction. Three layers are shown, representing simple, complex and bipole cells.

simple cells. Hypercomplex cells perform spatial and directional competition (inhibition) for edge formation. Bipole cells perform long-range cooperation for edge enhancement, and exert positive feedback (excitation) onto the hypercomplex cells. Our present implementation does not include the FCS model, which completes and fills features through diffusive spatial filtering of the image gated by the edges formed in the BCS.

We adopted the BCS algorithm for analog continuous-time implementation on a hexagonal grid, extending in three directions x, y and z on the focal plane as indicated schematically in Figure 23. For notational convenience, let subscript 0 denote the center pixel and $\pm x$, $\pm y$ and $\pm z$ its six neighbors. Components of each complex cell "vector" $\mathbf{C}_i$, along three directions of edge selectivity, are indicated with superscript indices x, y and z.

To facilitate testing the basic cells comprise of a photosensor sourcing a current indicating light intensity, a normalizing diffusion network for the intensity currents, gradient computation nodes, and one pseudo-complex cell and bipole cell for each of the three directions.

The photosensors generate a current I_i which is normalized through a diffusive network [Andreou *et al.*, 1995]. Through current mirrors, the currents I_i propagate in the three directions x, y, and z as noted in Figure 23. Rectified finite-difference gradient estimates of I_i are obtained for each of the three hexagonal directions. These gradients excite the complex cells C_i^j.

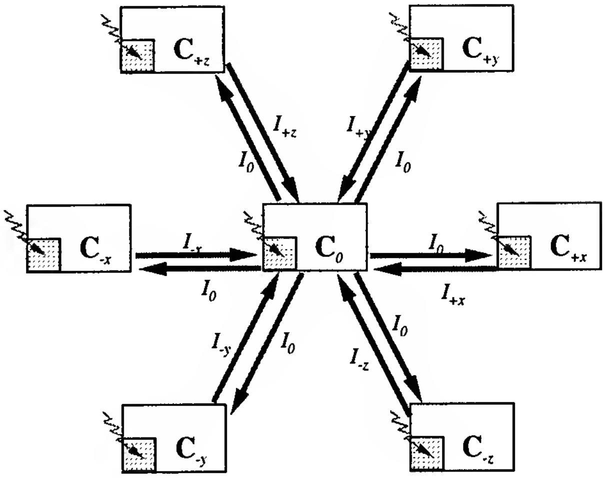


Figure 23: Hexagonal arrangement of BCS pixels, at the level of simple and complex cells, extending in three directions x, y and z in the focal plane.

Lateral inhibition among spatially (i) and directionally (j) neighboring complex cells implements the function of hypercomplex cells for edge sharpening. The complex output (C_i^j) is inhibited by local complex cell outputs in the two competing directions of j. C_0 is additionally inhibited by complex cells of the four nearest neighbors in competing locations i with parallel orientation.

A directionally selective interconnected diffusive network of bipole cells B_i^j provides long range cooperative feedback, and strengthens edges and curves while reducing false edges. C_i^j is excited by bipole interaction received from the bipole cell B_i^j on the line crossing i in the same direction j.

The equations describing the operation of the (hyper-)complex cells in our circuit implementation are as follows:

$$\begin{aligned} C_0^x &= 3D\,|I_z + I_y - I_{-y} - I_{-z}| - \alpha(C_0^y + C_0^z) \\ &\quad -\alpha'(C_z^x + C_y^x + C_{-z}^x + C_{-y}^x) + \beta B_0^x \quad (9) \end{aligned}$$

where:

1. $|I_z + I_y - I_{-y} - I_{-z}|$ is the rectified gradient input from the complex cells;

2. $\alpha(C_0^y + C_0^z)$ is the inhibition by local opposing directions;

3. $\alpha'(C_z^x + C_y^x + C_{-z}^x + X_{-y}^x)$ is inhibition from non-aligned neighbors in the same direction; and

4. βB_0^x is the excitation of long-range cooperation by the bipole cell.

The constants α, α' and β are set independently by externally applied bias voltages.

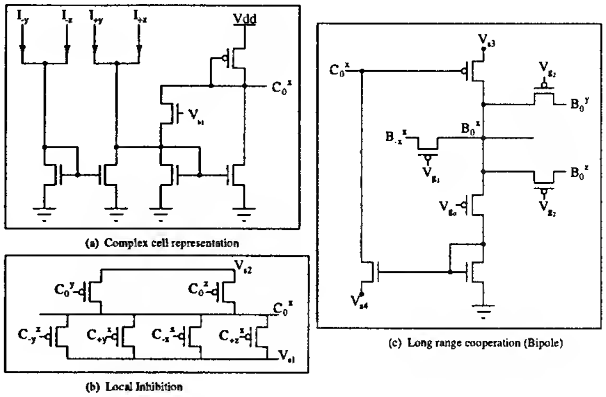
The bipole cell resistive grid implements a three-fold directionally polarized long-range diffusive kernel, basically as follows:

$$B_0^x = 3DK_xC_0^x + K_yC_0^y + K_zC_0^z \tag{10}$$

where K_x, K_y, and K_z represent identical but rotated bipole kernels with polarization in the x, y and z directions. The kernels are implemented by three linear networks of diffusor elements [Andreou *et al.*, 1995; Boahen and Andreou, 1992] complemented with cross-links of adjustable strength to control the degree of direction selectivity, besides the spatial spread of the kernel. Finally, the result (2) is locally normalized using current-mode circuitry, before it is fed back onto the complex cells.

The simplified circuit diagram of the BCS cell, including complex and bipole cell functions on a hexagonal grid, is shown in Figure 24. The complex cell portion in Figure 24A combines intensities I_y, I_{-y}, I_z, and I_{-z} received from neighboring cells to compute the rectified gradient in (1), using standard current mirrors and an absolute value circuit. A pMOS load converts the complex cell output into a voltage representation C_0^x for distribution to neighboring nodes and complementary orientations: local inhibition for spatial and directional competition in Figure 24B, and long-range cooperation through the bipole layer in Figure 24C.

Voltage biases control the spatial extent and directional selectivity of the interactions, as well as the relative strength of inhibition and excitation, and the level of renormalization, for the complex and bipole cells. The values for g_0, g_1 and g_2 controlling the bipole kernel are set externally by applying gate bias voltages V_{g0}, V_{g1} and V_{g2}, respectively. Likewise, the constants α, α' and β in (1) are set independently by the applied source voltages V_{s1}, V_{s2} and V_{s3}. Normalization of the bipole response for improved stability of edge formation is achieved by modulating V_{s3} through an additional diffusive network that acts as a localized Gilbert-type current normalizer (not shown in Figure 24).



(a) Complex cell representation

(b) Local Inhibition

(c) Long range cooperation (Bipole)

Figure 24: Simplified circuit schematic of one BCS cell in the hexagonal array. (A) Complex cell rectified gradient calculation. (B) Hypercomplex cell spatial and orientational inhibition. (C) Bipole cell directional long range cooperation.

A prototype 12 × 12 pixel array has been fabricated through MOSIS. The prototype pixel unit has been designed for optimal testability, and has not been optimized for density. The pixel contains 86 transistors including a phototransistor (for evaluation purposes), a large pMOS sample-and-hold capacitor, and three networks of interconnections in each of the three directions, requiring a fan-in/fan-out of 18 currents at the interface of each pixel unit. Testing has already provided key insights on further improvements in the architecture.

We expect this BCS chip to offer an important tool to further understand the operation and dynamics of BCS and related algorithms. In future research, we plan to extend the design to incorporate the Feature Contour System

(FCS), and scale up the dimension of the array to an appropriate size for smart vision and pattern recognition systems, using state of the art CMOS technology. Based on the current design, a 10,000-pixel array in 0.5 μm CMOS technology would fit a 1 cm^2 die.

11.4 Chip to Chip Communication

Ultimately, the prototype of the BCS/FCS will run interactively a computer that handles loading/unloading of the special chips and computes stages for which chips are not currently developed. Currently, our raw SAR data is contained in flat files that that is converted into corresponding analog data and loaded into Stage 1. Each chips will be serviced asynchronous spatial oversampling data converters that unloads the chip and send the data to the next location. In the case of rotating spatial filters, we have the choice of either rotating the filters electronically or rotating the data. As a prototype, the latter seems preferable. Locations can be encoded in a linear or nonlinear mapping that will be assigned with Read Only Memory (ROM) Chips. Preliminary work in this direction is both exciting and promising.

11.5 VLSI Architectures for the ART Family of Learning Machines

In this subtask of the project we focus on intelligent memories based on the ART family of learning machines and leveraging state of the art fabrication technologies for digital, analog and multilevel floating gate storage.

The general chip architecture for memory based processors is shown in Figure 25. This is a general floorplan applicable to ART1, fuzzy ART, and ARTMAP network models. The central array contains a very dense matrix of the synaptic cells. These can be analog with continuous updated [Cohen, Abshire, and Cauwenberghs, 1997], digital [Pouliquen, Andreou, and Strohbehn, 1997; Serrano-Gotarredona and Linares-Barranco, 1997]. The bottom cells are encharged of providing the appropriate biasing to the synaptic cells. This biasing depends on the input pattern. The top cells function is to implement the learning rule. Every time a row wins the competition and is selected for recode, its weight vector is copied to the top cells. This vector is then updated taking into account the actual input pattern, and stored back into the winning row. The right cells perform the vigilance criterion and the rest of computations needed to obtain the choice functions for the Winner-Take-All. The Winner-Take-All selects the maximum and its address, properly encoded, is sent to the outside of the chip.

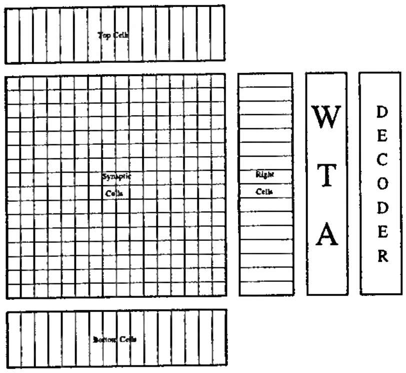


Figure 25: VLSI floorplan for ART network processors.

11.6 Multi-Chip ART1 Systems

Crucial in any large scale deployment of memory based processors is a scalable design where each individual chip has appropriate hooks to interface to similar chips in a multi-chip module. We are exploring multi-chip architecture issues using the ART1 memory based processor chip developed at the National Microelectronics Center in Sevilla, Spain [Serrano-Gotarredona and Linares-Barranco, 1997]. This chip is an improved version of the first ART1 prototype reported by Serrano-Gotarrendona and Linares-Barranco [1996] which improves manufacturing yield from 6% to 98% besides some minor problems encountered in the first design. This chip implements an ART1 network with a 50 nodes ART1 layer and a 10 F2 layer. 10 chips are available. These chips can be assembled in an

N by M array so that an ART1 with Nx50 F1 nodes and Mx10 F2 nodes can be assembled. Some preliminary multichip prototypes have been tested already that implement both ART1 and ARTMAP systems [Serrano-Gotarredona and Linares-Barranco, 1997]. Both Mrs. Teresa Serrano-Gotarredona and Dr. Bernabe Linares-Barranco are visiting Johns Hopkins during the 1996–1997 academic year on a Fulbright scholarship and postdoctoral fellowship respectively.

One major problem when increasing the number of nodes in the F2 layer is making the Winner-Take-All to preserve its precision when it is split among several chips. A Winner-Take-All circuit requires a certain degree of precision so that all of its inputs receive the same treatment. This is difficult to achieve in practice with high precision. The reason is that technological parameters of transistors suffer very large changes from chip to chip, and this would make that Winner-Take-All inputs of one chip would receive a different treatment that Winner-Take-All inputs from another chip. In the present prototype no special Winner-Take-All design to overcome this problem has been included. However, some efforts towards this goal have been already considered [Serrano-Gotarredona and Linares-Barranco, 1997]. The idea is to reformulate the Winner-Take-All operation in current domain. This allows to replicate and compare currents locally in one chip and transport currents from chip to chip. As long as current replication and comparison is maintained precise locally on each chip overall multichip WTA precision will be preserved.

We have designed a PCI based co-processor board and software under Linux and X-windows. The board has multiple ART1 chips so that ART1 systems of Nx50 F1 nodes and Mx10 F2 nodes (with N+M¡=3D10) can be realized. Based on the experience of the Sevilla group and our experience with intelligent memory designs we are architecting the next generation of ART family chips with digital storage and analog processing are also been prototyped in state of the art CMOS process.

12 References

Andreou, A.G., Meitzler, R.C., Strohbehn, K., and Boahen, K.A. (1995). Analog VLSI neuromorphic image acquisition and pre-processing systems. *Neural Networks*, **8**, 1323–1347.

Bachelder, I.A. and Waxman, A.M. (1994). Mobile robot visual mapping and localization: A view-based neurocomputational architecture that emulates hippocampal place learning. *Neural Networks*, **7**, 1083–1099.

Bachelder, I.A., Waxman, A.M., and Seibert, M. (1993). A neural system for mobile robot visual place learning and recognition. In **Proceedings of the world congress on neural networks (WCNN'93)**, **I**, 512–517. Hillsdale, NJ: Erlbaum Associates.

Baloch, A.A. and Grossberg, S. (1997). A neural model of high-level motion processing: Line motion and formotion dynamics. *Vision Research*, in press.

Baloch, A.A., Grossberg, S., Mingolla, E., and Nogueira, C.A.M. (1997). A neural model of first-order and second-order motion perception and magnocellular dynamics. **Technical Report CAS/CNS TR-96-030**, Boston University. Submitted for publication.

Baloch, A.A. and Waxman, A.M. (1991). Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN. *Neural Networks*, **4**, 271–302.

Baraldi, A. and Parmiggiani, F. (1995). A neural network for unsupervised categorization of multivalued input patterns: An application to satellite image clustering. *IEEE Transactions on Geoscience and Remote Sensing*, **33**, 305–316.

Barton, G. (1989). **Elements of Green's functions and propagation**. Oxford: Clarendon Press.

Bernardon, A.M. and Carrick, J.E. (1995). A neural system for automatic target learning and recognition applied to bare and camouflaged SAR targets. *Neural Networks*, **8**, 1103–1108.

Boahen, K.A. and Andreou, A.G. (1992). A contrast sensitive silicon retina with reciprocal synapses. In **Advances in neural information processing systems, Volume 4**. San Mateo, CA: Morgan Kaufmann Publishers, pp. 764–772. Reprinted in C. Koch (Ed.), **Analog vision chips**, IEEE Press, 1993.

Bonmassar, G. and Schwartz, E.L. (1996a). Lie groups, space-variant fourier analysis and the exponential chirp transform. In *Computer Vision and*

Pattern Recognition, in press.

Bonmassar, G. and Schwartz, E.L. (1996b). Fourier analysis and cortical architectures: the exponential chirp transform. *Real-Time Vision*, in press.

Bonmassar, G. and Schwartz, E.L. (1996c). Space-variant fourier analysis: the exponential chirp transform. *IEEE Pattern Analysis and Machine Vision*, in press.

Borden, B. (1993). Problems in airborne target recognition. *Inverse Problems*, **10**, 1009–1022.

Bradski, G. and Grossberg, S. (1995). Fast-learning VIEWNET architecture for recognizing three-dimensional objects from multiple two-dimensional views. *Neural Networks*, **8**, 1053–1080.

Bregman, A.L. (1981). Asking the "what for" question in auditory perception. In M. Kubovy and J.R. Pomerantz (Eds.), **Perceptual organization**. Hillsdale, NJ: Erlbaum Associates, pp. 99–118.

Brodatz, P. (1966). **Textures**. New York: Dover.

Burke, L.I. (1994). Neural methods of the traveling salesman problem: Insights from operations research. *Neural Networks*, **7**, 681–690.

Carpenter, G.A. (1997). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks*, in press.

Carpenter, G.A., Gjaja, M.N., Gopal, S., and Woodcock, C. E. (1997). ART neural networks for remote sensing: Vegetation classification from Landsat TM and terrain data. *IEEE Transactions on Geoscience and Remote Sensing*, in press.

Carpenter, G.A. and Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115.

Carpenter, G.A. and Grossberg, S. (1987b). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26**, 4919–4930.

Carpenter, G.A. and Grossberg, S. (1991). **Pattern recognition by self-organizing neural networks**. Cambridge, MA: MIT Press.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D.B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, **3**, 698–713.

Carpenter, G.A., Grossberg, S., and Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, **4**, 565–588.

Carpenter, G. A., Grossberg, S. and Reynolds, J. H. (1995). A fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems. *IEEE Transactions on Neural Networks*, **6**, 1330–1336.

Carpenter, G.A., Grossberg, S., and Rosen, D.B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, **4**, 759–771.

Carpenter, G.A. and Markuzon, N. (1996). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. **Technical Report CAS/CNS TR-96-017**, Boston, MA: Boston University.

Carpenter, G.A. and Ross, W.D. (1993). ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. In **Proceedings of the world congress on neural networks (WCNN'94)**, **III**, 649–656. Hillsdale, NJ: Erlbaum Associates.

Carpenter, G.A. and Ross, W.D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, **6**, 805–818.

Caudell, T.P. and Healy, M.J. (1994). Adaptive Resonance Theory networks in the Encephalon autonomous vision system. In**Proceedings of the 1994 IEEE international conference on neural networks**, **II**, 1235–1240. Piscataway, NJ: IEEE.

Caudell, T.P., Smith, S.D.G., Escobedo, R., and Anderson, M. (1994). NIRS: Large scale ART 1 neural architectures for engineering design retrieval. *Neural Networks*, **7**, 1339–1350.

Chey, J., Grossberg, S., and Mingolla, E. (1997a). Neural dynamics of motion processing and speed discrimination. *Vision Research*, in press.

Chey, J., Grossberg, S., and Mingolla, E. (1997b). Neural dynamics of motion grouping: From aperture ambiguity to object speed and direction. *Journal of the Acoustical Society of America*, in press.

Choi, J., Ly, S., Healy, M., and Smith, S. (1996). Prediction of cutter condition using LAPART. In **Proceedings of the international conference on neural networks: Plenary, panel, and special sessions**, pp. 226–230.

Christodoulou, C.G., Huang, J., Georgiopoulos, M., and Liou, J.J. (1995). Design of gratings and frequency selective surfaces using fuzzy ARTMAP neural networks. *Journal of Electromagnetic Waves and Applications*, **9**, 17–36.

Cohen, M., Abshire, P., and Cauwenberghs, G. (1997). Mixed-mode VLSI architecture implementing fuzzy ART. In **Proceedings of the 31st annual conference on information sciences and systems**, in press.

Cohen, M.A. and Grossberg, S. (1984). Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance. *Perception and Psychophysics*, **36**, 428–456.

Cunningham, R.K., Beck, J., and Mingolla, E. (1996). Visual search for a foreground object in continuous, naturalistic displays: The importance of shadows and occlusion. *Investigative Ophthalmology and Visual Science*, **37**(3), S299.

Dempster, A.P., Larid, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, **39**, 1–38.

Deterding, D.H. (1989). **Speaker normalisation for automatic speech recognition**. PhD Thesis, University of Cambridge.

Dubrawski, A. and Crowley, J.L. (1994a). Learning locomotion reflexes: A self-supervised neural system for a mobile robot. *Robotics and Autonomous Systems*, **12**, 133–142.

Dubrawski, A. and Crowley, J.L. (1994b). Self-supervised neural system for reactive navigation. In **Proceedings of the IEEE international conference on robotics and automation**, pp. 2076–2081. Los Alamitos, CA: IEEE Computer Society Press.

El-Fallah, A.I. and Ford, G.E. (1994). Nonlinear adaptive image filtering based on inhomogeneous diffusion and differential geometry. *SPIE Image and Video Processing II*, **2182**, 49–63.

Engel, G., Greve, D., Lubin, J., and Schwartz, E.L. (1994). Space-variant active vision and visually guided robotics: Design and construction of a high-performance miniature vehicle. In **Proceedings of the international conference on pattern recognition**, ICPR-12.

Feng, C., Sutherland, A., King, S., Muggleton, S., and Henery, R. (1993). Symbolic classifiers: Conditions to have good accuracy performance. *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pp. 371–380.

Fischl, B. (1996). **Learning nonlinear filtering, anisotropic diffusion and space variant vision.** PhD Thesis, Department of Cognitive and Neural Systems, Boston University.

Fischl, B., Cohen, M.A., and Schwartz, E.L. (1997a). The local structure of space-variant images. *Neural Networks*, in press.

Fischl, B., Cohen, M.A., and Schwartz, E.L. (1997b). Real-time anisotropic diffusion using space-variant vision. *International Journal of Computer Vision*, in press.

Fischl, B. and Schwartz, E.L. (1996). Adaptive nonlinear filtering for non-linear anisotropic diffusion approximation in image processing. In **Proceedings of the international conference on pattern recognition (ICPR)**, in press.

Fischl, B. and Schwartz, E.L. (1997a). Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press.

Fischl, B. and Schwartz, E.L. (1997b). Adaptive non-local filtering: A fast alternative to anisotropic diffusion for image enhancement. **Technical Report CAS/CNS TR-96-033**, Boston University.

Fischl, B. and Schwartz, E.L. (1997c). Fast adaptive alternatives to non-linear diffusion in image enhancement: Greens function approximators and non-local filters. In **First international conference on scale space theory in computer vision**, in press.

Francis, G. and Grossberg, S. (1996). Cortical dynamics of form and motion integration: Persistence, apparent motion, and illusory contours. *Vision Research*, **36**, 149–173.

Frey, P.W. and Slate, D.J. (1991). Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, **6**, 161–182.

Gan, K.W. and Lua, K.T. (1992). Chinese character classification using an adaptive resonance network. *Pattern Recognition*, **25**, 877–888.

Gaudiano, P., Zalama, E., and López-Coronado, J. (1996). An unsupervised neural network for low-level control of a mobile robot: Noise resistance, stability, and hardware implementation. *IEEE Transactions on Systems, Man, and Cybernetics*, **26**, 485–496.

Ghahramani, Z. and Jordan, M.I. (1994). Learning from incomplete data. *A.I. Memo No. 1509 and C.B.C.L. Paper No. 108*, Massachusetts Institute of

Technology.

Gjerdingen, R.O. (1990). Categorization of musical patterns by self-organizing neuronlike networks. *Music Perception*, **7**, 339–370.

Gopal, S., Sklarew, D.M., and Lambin, E. (1994). Fuzzy-neural networks in multi-temporal classification of landcover change in the Sahel. In **Proceedings of the DOSES workshop on new tools for spatial analysis**. Lisbon, Portugal, DOSES, EUROSTAT. ECSC-EC-EAEC: Brussels, Luxembourg, pp. 55–68.

Greenspan, H. (1996). Non-parametric texture learning. In **Early visual learning**, S. Nayar and T. Poggio (Eds.). New York: Oxford University Press.

Greenspan, H., Goodman, R., Chellappa, R., and Anderson, C.H. (1994). Learning texture discrimination rules in a multiresolution system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 894–901.

Greve, D., Engel, G., and Schwartz, E.L. (1997). Instrumentation and control of a miniature pan-tilt actuator: The spherical pointing motor. *IEEE Robotics and Automation*, submitted for publication.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, **23**, 187–202.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, **87**, 1–51.

Grossberg, S. (1982). The processing of expected and unexpected events during conditioning and attention: A psychophysiological theory. *Psychological Review*, **89**, 529–572.

Grossberg, S. (Ed.) (1987). **The adaptive brain, Volume I**. Amsterdam: North-Holland.

Grossberg, S. (1994). 3-D vision and figure-ground separation by visual cortex. *Perception and Psychophysics*, **55**, 48–120.

Grossberg, S. (1997). How is a moving target continuously tracked behind occluding cover? In T. Watanabe (Ed.), **High level motion processing**. Cambridge, MA: MIT Press, in press.

Grossberg, S. and McLoughlin, N.P. (1997). Cortical dynamics of 3-D surface perception: Binocular and half-occluded scenic images. **Technical Report CAS/CNS TR-95-022**, Boston University. Submitted for publication.

Grossberg, S., Mingolla, E., and Ross, W.D. (1994). A neural theory of attentive visual search: Interactions of boundary, surface, spatial, and object representations. *Psychological Review*, **101**, 470–489.

Grossberg, S., Mingolla, E., and Ross, W.D. (1997). Visual brain and visual perception: How does the cortex do perceptual grouping? *Trends in Neurosciences*, in press.

Grossberg, S., Mingolla, E., and Todorović, D. (1989). A neural network architecture for preattentive vision. *IEEE Transactions on Biomedical Engineering*, **36**, 65–84.

Grossberg, S., Mingolla, E., and Williamson, J.R. (1995). Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation. *Neural Networks*, **8**, 1005–1028.

Grossberg, S. and Pessoa, L. (1997). Texture segregation, surface representation, and figure-ground separation. **Technical Report CAS/CNS TR-96-025**, Boston University. Submitted for publication.

Grossberg, S., Roberts, K., Aguilar, M., and Bullock, D. (1997). A neural model of multimodal adaptive saccadic eye movement control by superior colliculus. **Technical Report CAS/CNS TR-96-029**, Boston University. Submitted for publication.

Grossberg, S. and Todorović, D. (1988). Neural dynamics of 1-D and 2-D brightness perception: A unified model of classical and recent phenomena. *Perception and Psychophysics*, **43**, 241–277.

Grossberg, S. and Williamson, J.R. (1997). A self-organizing neural system for learning to recognize textured scenes. **Boston University Technical Report CAS/CNS TR-97-001**.

Grossberg, S.G., Rubin, M.A., and Streilein, W.W. (1996). Buffered reset leads to improved compression in fuzzy ARTMAP classification of radar range profiles. In C.H. Dagli *et al.* (Eds.), **Intelligent engineering systems through artificial neural networks, 6. Smart engineering systems: Neural networks, fuzzy logic and evolutionary programming**. New York: ASME Press. pp. 419–424.

Ham, F.M. and Han, S.W. (1993). Quantitative study of the QRS complex using fuzzy ARTMAP and the MIT/BIH arrhythmia database. In **Proceedings of the world congress on neural networks (WCNN'93), I**, 207–211. Hillsdale, NJ: Erlbaum Associates.

Hinck, T. and Hubbard, A. (1996). Simulated retinal center/surround artificial neuroprocessing using analog VLSI. In **Proceedings of the 1996 computational neuroscience conference**, Boston.

Hoffer, R.M. and Staff (1975). Natural resources mapping in mountainous terrain by computer analysis of ERTS-1 satellite data. *Agricultural Experiment Station Research Bulletin 919* and *LARS Contract Report 061575*, West Lafayette, IN: Purdue University.

Hudson, S. and Psaltis, D. (1993). Correlation filters for aircraft identification from radar range profiles. *IEEE Transactions on Aerospace and Electronic Systems*, **29**, 741–748.

Hummel, A. (1986). Representations based on zero-crossing in scale-space. In M. Fischler and O. Firschein (Eds.), **Readings in computer vision: Issues, problems, principles, and paradigms**. Los Angeles: Morgan Kaufmann Publishers.

Johnson, C. (1993). Agent learns user's behavior. *Electrical Engineering Times*, June 28, pp. 43, 46.

Kalkunte, S.S., Kumar, J.M., and Patnaik, L.M. (1992). A neural network approach for high resolution fault diagnosis in digital circuits. In **Proceedings of the international joint conference on neural networks, I**, 83–88. Piscataway, NJ: IEEE.

Kanizsa, G. (1979). **Organization in vision: Essays in Gestalt perception**. New York: Praeger Press.

Kasperkiewicz, J., Racz, J., and Dubrawski, A. (1995). HPC strength prediction using artificial neural network. *Journal of Computing in Civil Engineering*, **9**, 279–284.

Kim, J.W., Jung, K.C., Kim, S.K., and Kim, H.J. (1995). Shape classification of on-line Chinese character strokes using ART 1 neural network. In **Proceedings of the world congress on neural networks (WCNN'95), II**, 191–194. Hillsdale, NJ: Erlbaum Associates.

Koch, M.W., Moya, M.M., Hostetler, L.D., and Fogler, R.J. (1995). Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition. *Neural Networks*, **8**, 1081–1102.

Koenderink, J. (1984). The structure of images. *Biological Cybernetics*, **50**, 363–370.

Ly, S. and Choi, J.J. (1994). Drill condition monitoring using ART 1. In **Proceedings of the 1994 IEEE international conference on neural networks, II**, 1226–1229. Piscataway, NJ: IEEE.

MacLeod, K.J. and Surkan, A.J. (1992). Algorithm performance in ART 1-like clustering of descriptor subsets for document retrieval. In **Proceedings of the international joint conference on neural networks, I**, 685–689. Piscataway, NJ: IEEE.

McLoughlin, N.P. and Grossberg, S. (1997). Cortical computation of stereo disparity. *Vision Research*, in press.

Mead, C.A. (1989). **Analog VLSI and electron neural systems**. Reading, MA: Addison-Wesley.

Murphy, P.M. and Aha, D.W. (1992). UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [machine-readable data repository]

Murshed, N.A., Bortolozzi, F., and Sabourin, R. (1995). Off-line signature verification, without a priori knowledge of class ω_2: A new approach. In **Proceedings of ICDAR 95: The third international conference on document analysis and recognition.**

Murshed, N.A., Bortolozzi, F., and Sabourin, R. (1996). A fuzzy ARTMAP-based classification system for detecting cancerous cells, based on the one-class problem approach. In **Proceedings of the 13th international conference on pattern recognition (ICPR'96)**.

Novak, L., Burl, M., Chaney, R., and Owirka, G. (1990). Optimal processing of polarimetric synthetic-aperture radar imagery. *The Lincoln Laboratory Journal*, **3**, 273–290.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629–639.

Pouliquen, P.O., Andreou, A.G., and Strohbehn, K. (1997). Winner-take-all associative memory: A Hamming distance vector quantizer. *Journal of Analog Integrated Circuits and Signal Processing*, **13**, 103–114.

Racz, J. and Dubrawski, A. (1995). Artificial neural network for mobile robot topological localization. *Robotics and Autonomous Systems*, **16**, 73–80.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–408. Reprinted in J.A. Anderson and E. Rosenfeld (Eds.), **Neurocomputing: Foundations of research**. Cambridge, MA: MIT Press, 1988, pp. 18–27.

Rosenblatt, F. (1962). **Principles of neurody-**

namics. Washington, DC: Spartan Books.

Rubin, M.A. (1995). Application of fuzzy ARTMAP and ART-EMAP to automatic target recognition using radar range profiles. *Neural Networks*, **8**, 1109–1116.

Ruda, H. and Snorrason, M. (1996). Automated construction of a hierarchy of self-organized neural network classifiers. In **Proceedings of the Southeastern Simulation Conference**.

Rumelhart, D.E., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), **Parallel distributed processing**. Cambridge, MA: MIT Press, pp. 318–362.

Seibert, M. and Waxman, A.M. (1992). Adaptive 3D object recognition from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, 107–124.

Seibert, M. and Waxman, A.M. (1993). An approach to face recognition using saliency maps and caricatures. In **Proceedings of the world congress on neural networks (WCNN'93), III**, 661–664. Hillsdale, NJ: Erlbaum Associates.

Serrano-Gotarredona, T. and Linares-Barranco, B. (1996). A real-time clustering microchip neural engine. *IEEE Transactions on VLSI Systems*, 4(2), 195–209.

Serrano-Gotarredona, T. and Linares-Barranco, B. (1997). An ART1 microchip and its use in multi-ART1 systems. *IEEE Transactions on Neural Networks*, in press.

Smith, J.W. (1962). ADAP II, an adaptive routine for the LARC computer. Navy Management Office, September 1962 (available through the Logistics Management Institute Library).

Smith, C.R. and Goggans, P.M. (1992). Radar target identification. *IEEE Antennas and Propagation Magazine*, **35**, 27–38.

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., and Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In **Proceedings of the symposium on computer applications and medical care**. IEEE Computer Society Press, pp. 261–265.

Soliz, P. and Caudell, T.P. (1996). Inferring future states of the atmosphere with a laterally primed adaptive resonance theory (LAPART) neural network. In **Proceedings of the world congress on neural networks (WCNN'96)**, 822–825. Hillsdale, NJ: Erlbaum Associates.

Soliz, P. and Donohoe, G.W. (1996). Adaptive resonance theory neural network for fundus image segmentation. In **Proceedings of the world congress on neural networks (WCNN'96)**, 1180–1183. Hillsdale, NJ: Erlbaum Associates.

Srinivasa, N. and Sharma, R. (1996). A self-organizing invertible map for active vision applications. In **Proceedings of the world congress on neural networks (WCNN'96)**, 121–124. Hillsdale, NJ: Erlbaum Associates.

Strahler, A.H., Logan, T.L., and Bryant, N.A. (1978). Improving forest cover classification accuracy from Landsat by incorporating topographic information. In **Proceedings of the 12th international symposium on remote sensing of the environment**. Ann Arbor, MI: Environmental Research Institute of Michigan, pp. 927–942.

Suzuki, Y., Abe, Y., and Ono, K. (1993). Self-organizing QRS wave recognition system in ECG using ART 2. In **Proceedings of the world congress on neural networks (WCNN'93), IV**, 39–42. Hillsdale, NJ: Erlbaum Associates.

Tarng, Y.S., Li, T.C., and Chen, M.C. (1994) Tool failure monitoring for drilling processes. In **Proceedings of the 3rd international conference on fuzzy logic, neural nets, and soft computing**. Iizuka, Japan, pp. 109–111.

ter Haar Romeny, B.M. (1994). **Geometry driven diffusion in computer vision**. Monterey, CA: Kluwer.

Tse, P., Cavanagh, P., and Nakayama, K. (1997).The role of parsing in high level motion processing. In T. Watanabe (Ed.), **High level motion processing**. Cambridge, MA: MIT Press, in press.

Tse, P. and Wang, D.D. (1996). A hybrid neural networks based machine condition forecaster and classifier by using multiple vibration parameters. In **Proceedings of the 1994 IEEE international conference on neural networks, IV**, 2096–2100. Piscataway, NJ: IEEE.

U.S. Army Research Office (1994). *Report of the Working Group on Automatic Target Recognition*, 20 September.

Varma, A., Woods, W.H. III, and Agogino, A. (1996). A machine learning approach to automated design classification, association and retrieval. In J. Gero and F. Sudweeks (Eds.), **Artificial intelligence in design**. The Netherlands: Kluwer Academic Publishers, pp. 429–445.

Volakis, J.L. (1994). XPATCH: A high-frequency electromagnetic-scattering prediction code and environment for complex three-dimensional objects. *IEEE Antennas and Propagation Magazine*, **36**, 65–69.

Wang, T., Xu, Q., and Ziaoliang, X. (1992). Character recognition through improving adaptive resonance theory (ART). In **Proceedings of the international joint conference on neural networks**, **I**, 761–764. Piscataway, NJ: IEEE.

Waskiewicz, J. and Cauwenberghs, G. (1997). Focal-plane analog VLSI implementation of the BCS image segmentation algorithm. In **Proceedings of the 31st annual conference on information sciences and systems**, in press.

Waxman, A.M., Gove, A.N., Fay, D.A., Racamato, J.P., Carrick, J.E., Seibert, M.C., and Savoye, E.D. (1997). Color night vision: Opponent processing in the fusion of visible and IR imagery. *Neural Networks*, **10**(1), 1–6.

Waxman, A.M., Seibert, M., Bernardon, A.M., and Fay, D.A. (1993). Neural systems for automatic target learning and recognition. *The Lincoln Laboratory Journal*, **6**(1), 77-116.

Waxman, A.M., Seibert, M.C., Gove, A., Fay, D.A., Bernardon, A.M., Lazott, C., Steele, W.R., and Cunningham, R.K. (1995). Neural processing of targets in visible, multispectral IR and SAR imagery. *Neural Networks*, **8**, 1029–1051.

Werbos, P. (1974). **Beyond regression: New tools for prediction and analysis in the behavioral sciences**. PhD Thesis, Cambridge, MA: Harvard University.

Whiteley, J.R., Davis, J.F., Mehrotra, A., and Ahalt, S.C. (1996). Observations and problems applying ART 2 for dynamic sensor pattern interpretation. *IEEE Transactions on Neural Networks*, **26**, 423–437.

Wienke, D. (1993). ART pattern recognition software for chemists. User Documentation, University of Nijmegen.

Wienke, D. (1994). Neural resonance and adaption: Towards nature's principles in artificial pattern recognition. In L. Buydens and W. Melssen (Eds.), **Chemometrics: Exploring and exploiting chemical information**. Nijmegen, NL: University Press.

Wienke, D. and Kateman, G. (1994). Adaptive Resonance Theory based artificial neural networks for treatment of open-category problems in chemical pattern recognition: Application to UV-Vis and IR spectroscopy. *Chemometrics and Intelligent Laboratory Systems.*

Wienke, D., Xie, Y., and Hopke, P.K. (1994). An adaptive resonance theory based artificial neural network (ART 2A) for rapid identification of airborne particle shapes from their scanning electron microscopy images. *Chemometrics and Intelligent Laboratory Systems.*

Williamson, J.R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, **9**, 881–897.

Williamson, J.R. (1997). A constructive, incremental-learning network for mixture modeling and classification. *Neural Computation*, in press.